Remote High-Definition Visual Monitoring of Cetaceans from offshore vessels and platforms

Baruwa, Ladipo

Award date:
2017

Awarding institution:
University of Bath

UNIVERSITY OF BATH

Remote High-Definition Visual Monitoring of Cetaceans from offshore vessels and platforms

by
Abdulquadir Ladipo Baruwa

for the degree of

Doctor of Philosophy

Department of Electronics and Electrical Engineering

August 2017

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr Adrian Evans, for all his guidance and support during my degree. His advice and clear comments have been immensely beneficial and greatly appreciated. Also, a very special thanks to my second supervisor, Dr Robert Watson, for all his brilliant ideas and comments.

This work would have not been possible without the financial support from Seiche Ltd who sponsored my PhD studies and several data collection/research activities associated with this work. I am especially indebted to Mr Roy Wyatt, CEO/MD and Mr Simon Cole, Director at Seiche, who have been immensely supportive throughout this period. I am grateful to all my colleagues and members of staff at Seiche, whom I have had the pleasure of working with during this and other related projects, especially Dr. Stephen Cook for helpful discussions and some proof reading.

This project started as an Innovate UK Knowledge Transfer Partnership (KTP) programme between Seiche Ltd and the University of Bath. Therefore, I would also like to thank the members of staff at the KTP office in University of Bath for all their kind advice during the early stages of this project.

Nobody has been more important to me in the pursuit of this project than members of my family. A very special thanks to my parents, my sisters and brothers for coping with me this last few months and their support throughout.

Most importantly, I wish to thank my loving and supportive wife, Sherry, who provide unending inspiration and unwavering support.

Abstract

This thesis introduces the Remote High-definition Visual Monitoring (RHVM) system. This system provides an affordable and sophisticated alternative to current methods of visual monitoring of cetaceans at sea.

There are several scenarios that require monitoring of marine mammals at sea; one of which includes efforts made to mitigate the effect of man-made noise (e.g. during seismic surveys) on the animals. Often used visual methods relies solely on humans (experts called Marine Mammal Observers) to detect the animals and subsequently estimate distance to them. In addition to problems caused by poor visibility at night, fog and fatigue, estimating distance at sea with the naked eyes is very difficult and is often guess work. Unsophisticated distance estimation methods, such as sighting stick, are not very accurate or precise and can result in unnecessary and expensive delays to the surveys or endanger animals.

These problems are addressed by combining the application of robust computer vision algorithms with relative cheap off the shelf sensors and the latest telecommunication system. The sea environment presents very peculiar challenges to computer vision methods due to constantly changing atmospheric conditions, unpredictable movement of the vessel and very cluttered image scenes due to the waves. A highly flexible multi-sensors system has been designed and tested.

A Real-time Automated Distance Estimation at Sea (RADES) algorithm has been developed for objective and recordable distance estimation at sea. The system tracks the vessel global orientation by detecting the horizon in images from a camera. Although horizon detection techniques have been studied in the past, the problem of real-time detection and ability to cope with a wide variety of conditions has not been effectively dealt with. The horizon detection technique developed here is made robust to weather effects by the application of the dark channel prior for pre-processing and robust to temporary occlusion by fusing visual measurement with inertial sensors data. Detailed mathematical analysis of the distance estimation technique is also given and a formula for estimating the resolution of the system is presented for the first time.

A new algorithm for Automated Recognition of Cetaceans at Sea (ARCS) has also been developed. The algorithm relies on a novel application of morphological operation that adapt to the content of the image scene for extraction of whales blows. The algorithm is capable of coping with a considerable amount of noise in the challenging sea environments; using Support Vector Machines (SVM) for classification. Steps towards training the SVM includes an effective data cleaning step based on Tomek Links and a scheme for dealing with a highly imbalanced data set is given.

Table of content

List of Abbreviations

AHRS	Attitude and Heading Reference System
ANN	Artificial Neural Networks
ARCS	Automated Recognition of Cetaceans at Sea
ASM	Active Shape Model
CMS	Camera monitoring system
DOF	Degrees of Freedom
DR	Distinguished regions
DWT	Discrete wavelet transform
EKF	Extended Kalman filter
FOV	Field of View
GUI	Graphical User Interface
HD	High-Definition
HMT	Hit-or-Miss Transform
IMU	Inertial Measuring Unit
INS	Inertial navigation system
IR	Infra-red
MMO	Marine Mammal Observers
MPA	Marine Protected Areas
MSER	Maximal Stable Extrema Region
NOAA	National Oceanic and Atmospheric Administration
PAM	Passive Acoustic Monitoring
PCA	Principle Component Analysis
PTU	Pan and Tilt Unit
QKF	Quaternion Kalman Filter
QP	Quadratic Programming
RADES	Real-time Automated Distance Estimation at Sea
RBF	Radial Bias Function
ROC	Receiver Operator Characteristic
RHVM	Remote High-definition Visual Monitoring
ROI	Region of interest
SFBS	Sequential floating backward
SFFS	Sequential floating forward selection
SOMP	Single Object Matching using Probing
SVM	Support Vector Machines
UAV	Unmanned air vehicle
UKF	Unscented Kalman filter
USV	Unmanned surface vehicles
VNC	Virtual network computing

Chapter 1 : Introduction

In this thesis, findings of research carried out and work done towards developing a Remote High-Definition Visual Monitoring (RHVM) system for monitoring of cetaceans at sea are described.

This chapter is an introduction to the thesis. In Section 1.1, a brief description of the problem that motivates this research is given and the main benefits of the system developed is given in Section 1.2. In Section 1.3, the main contributions of the thesis are described and the thesis outline is given.

1.1 Motivation: Marine Mammal Monitoring

There is a growing concern about the potential adverse effects of man-made activities that produce a significant amount of noise underwater on marine mammals. Research and studies have shown evidence of behavioural and physical effects [1]–[4]. Marine mammals depend on sound in their daily activities including navigation of the ocean, communication, finding prey and avoiding predators. This is because sound travels further underwater than light. As a result, marine mammals often modify their behaviour when exposed to intense noise [2]. Several other potential risks identified include physical injury from tissue damage, temporary and permanent noise-induced hearing loss, stranding (resulting in beaching), masking of echolocation signals and other indirect effects.

Pile driving, drilling and dredging operations associated with marine renewable energy development [5] and marine seismic surveys conducted by the oil and gas industry [4] are some of the man-made activities of concern. During seismic survey operations, for example, various forms of mitigation measures are implemented to ensure that marine mammals are safeguarded. In fact, the deployment of mitigation measures is a legal requirement in several parts of the world especially nations with high level of seismic activities including UK, USA, Brazil [6]. A standard form of mitigation involves defining an area surrounding the noise source (for example a radius of 500m around the source) called the mitigating zone. When a marine mammal is identified within this zone, mitigation actions are triggered; typically, a delay or suspension of operation.

One method of locating mammals in the mitigation zone called Passive Acoustic Monitoring (PAM) involves using an array of hydrophones to detect and localize marine mammal vocalizations. The drawback of this method is that it cannot detect marine mammals when they are not vocalizing; a typical behaviour when they are on the sea surface. Therefore, during daylight hours, visual monitoring of the mitigation zone by trained personnel (experts) called Marine Mammal Observers (MMO) is employed. While visual monitoring also has its drawbacks, since mammals can only be detected in daylight when they come to

the surface, PAM and MMOs are often employed alongside each other to increase the chances of locating marine mammals.

MMOs employ various means (discussed in Section 2.2) to detect marine mammals during visual monitoring. They continually search the mitigation area and when a marine mammal is identified on the surface, they must estimate its distance from the noise source, to determine whether it is in the mitigation zone. There is usually a line of communication between the MMOs and the seismic crew which allows MMOs to advise them when a marine mammal is detected in the mitigation zone so that mitigation actions can be triggered. The visual monitoring operation is typically carried out by one or more MMOs often located on the bridge (the topmost deck) of the vessel and other areas on deck with a clear view of the area around the vessel. The monitoring must be undertaken throughout the entire day when seismic operation is on-going. It is a very difficult operation because:

1. Marine mammals surface only for a few seconds most of the time
2. Fog, sea mist, swells etc. can severely impair visual detection of mammals
3. The human eye can only focus on a small area at a time
4. Estimating distance at sea with the naked eye is very difficult and is often guesswork

The unsophisticated methods often employed can result in partial (i.e. many false negative) detection of marine mammals and poor distance estimation of detected ones. In addition to the aforementioned difficulties, there is usually no way to capture evidence of marine mammal sightings or locations and so the operation staff must rely solely on the MMOs judgement. The need for a more sophisticated way of detecting cetaceans, recording sightings and estimating distance at sea is the problem that motivates this work.

1.2 Remote High-Definition Visual Monitoring (RHVM) system

Reliance on unsophisticated methods of visual monitoring for mitigation is not best practice, and this provides the motivation for the development of a new system. These are outlined as follows:

1. **Reduce cost and the risk of life involved in cetacean monitoring**
   Groups of two or three MMOs are often employed at a time to improve chances of detecting a marine mammal and they are rotated regularly throughout the offshore survey lifespan, sometimes using helicopters and/or chase boats. These are hazardous and risky operations. In addition, MMOs are exposed to various other hazards while carrying out visual monitoring for example slippery surfaces on deck due to spray.

2. **Improved protection of marine mammals.**
   The effect of underwater noise on marine mammals is still not well understood but there are indications that it could be fatal and quite distressful to the animals. Marine mammals only come to the surface for a few seconds most of the time and since the

human eye can only focus on one area at a time, they can be easily missed. This is exacerbated by the lapses of concentration and fatigue that can easily creep in when MMOs are carrying out their duties. The current method of visual monitoring is inherently error prone and human bias can result in marine mammals being wrongly judged to be inside/outside the mitigation zone.

3. **Evidence based system:**
   Offshore surveys cost a huge amount of money to carry out, even more so if they must stop operation due to the presence of marine mammals in the mitigation zone. As such, an inaccurate judgement that a marine mammal is in the mitigation zone can have significant cost implication. However, the current system of mitigation is inherently flawed because MMOs have hardly any evidence to back a claim that a marine mammal was in the mitigation zone, hence making them vulnerable to dismissal by unconvinced and reluctant operation crew.

4. **Flexible design and easily adaptable for different fields of use.**
   Seismic surveys are not the only source of underwater noise pollution. Also, there are other applications, including biological surveys, that require accurate distance estimation at sea and location of surface objects.

To address the above problems, the development of a Remote High-Definition Visual Monitoring (RHVM) system is proposed. The RHVM system is an evidence based system whereby sightings are recorded and positions of mammals relative to the exclusion zone can be made obvious by augmenting scene images with a graphic overlay of the mitigation zone and detected cetaceans.

The system addresses several issues including alleviating the risks often associated with visual monitoring by providing a means of monitoring from a relatively hazard-free area below deck or a remote location onshore via satellite links. It eliminates human bias by providing accurate and objective distance estimation on the sea surface within the limits of the resolution of the system. The system has been designed to cope with varying scenarios e.g. it takes advantage of measurements from multiple sensors when they are available but will still operate when only a single camera is available. In addition, it facilitates 24hr visual monitoring which is not possible with the current method.

As shown in Figure 1.1, the RHVM system consists of three mains parts: 1) a hardware sensing unit at the front end, 2) various software processing systems and 3) the monitoring system at the client end. The direction of the arrows shows the data flow from acquisition to display.

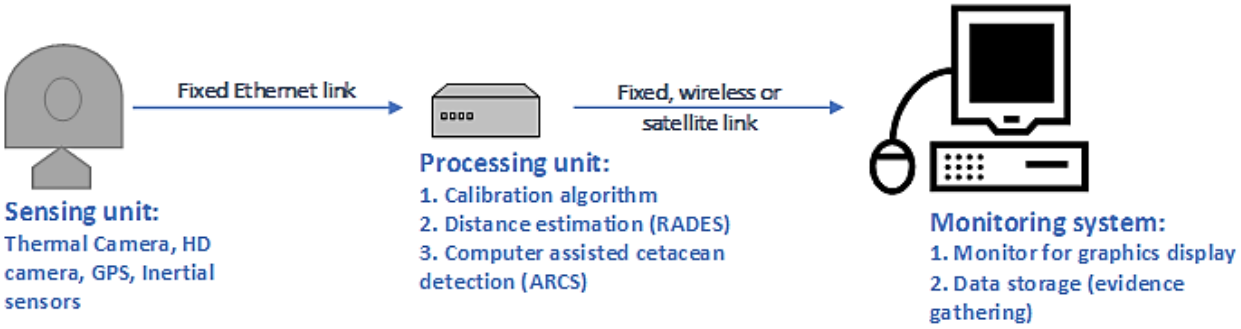


Figure 1.1: Block diagram of RHVM system

The main focus of the work in this thesis is on the various software algorithms that make up the processing system. However, the quality of data from the sensing unit and the way images are presented in the monitoring station is also addressed. The hardware units are designed using commercially available parts. The sensing unit and processing system are always located on the offshore platform whereas the monitoring station may be located at a remote location on the same platform, on another vessel or onshore; in which case, a wireless communication link is required e.g. over satellite.

1.3 Thesis Contributions and Outline

The main contributions of this thesis are of multiple fold; first is a new tool for accurate distance estimation at sea. Two methods are introduced, one based on trigonometry and the other based on the camera perspective model. Compared to previously reported work, all mathematical formulations are derived from first principles with no simplifying assumptions made. An analysis of the method is performed and as a result a method for estimating the resolution of the system is derived for the first time. This facilitates the selection of the correct camera lens parameters to achieve desired resolution in pixel/meter.

The distance estimation system relies on real-time detection of the horizon for attitude resolution. The system offers several advantages over existing image ranging solutions, most of which can only be used for offline processing where the horizon must be identified manually by an operator. They are designed mainly for biological surveys such as marine mammal population estimation surveys. The output of the system presented here is a real-time overlay of graphics on images for display in the monitoring system as shown in Figure 1.1; this augmented reality style approach is the first in this type of application.

A new computer vision system for horizon detection that is completely automated is proposed. The uniqueness of the horizon detection algorithm comes from the novel combination of several efficient image processing techniques to form a highly effective system capable of dealing with varying conditions at sea. In chapter 5, a selection of notable and well-established algorithms is reviewed. However, the specific problem of a real-time automated systems at sea capable of dealing with complex challenges caused by continually

changing atmospheric condition and noise due to waves has not been effectively dealt with in the literature. It was demonstrated that the algorithm presented here is capable of coping with these challenges by comparing it to well-known alternatives.

In addition, the algorithm offers several performance improvement over traditional methods including processing speed multiple times faster. It includes a metric that defines the confidence of the detection process, for use in Bayesian analysis. The well-known Kalman filter is incorporated to track the horizon and predict the position in the next image; this critical step facilitates real-time processing.

A second aspect of the horizon detection algorithm takes advantage of measurements from multiple sensors when they are available. An efficient quaternion Kalman filter is developed for tracking the horizon, incorporating measurements from multiple complementary sensor to provide the best estimates. This offers numerous performance benefits as well as making the system resilient to partial occlusion and noise. The sensor fusion filter relies on several calibration routines, which include simple but effective methods for estimating the relative orientation between the sensors and the temporal misalignment. The main benefit of these calibration methods is that they do not require any elaborate or complicated platform or rig. In addition, a new online camera calibration technique that does not use a calibration pattern is presented for use at sea. This affords additional benefits such as enabling camera lens parameters to change during operation (i.e. optical zoom and focus capability).

A new algorithm for automated recognition of cetacean features for computer assisted detection at sea is also proposed. Techniques that exist in the literature, as discussed in Chapters 2 and 6, require spatially varying thresholds. These techniques work best when sea conditions are constant. Here, a different approach to segment sea images is adopted, using a morphological technique that adapts to the changing scene and does not require a fixed threshold. This method allows segmentation and feature extraction (both in space and time) to be performed in a single operation. This method is shown to be able to cope effectively with changes in scene and environmental condition.

An analysis is performed to establish the whale blow characteristics based on the feature extraction step. This is then combined with an SVM classifier trained from a highly imbalanced data set. The imbalance is due to numerous clutter in the sea environment which is further exacerbated by the choice of sensor adopted here; that is a low cost and less sensitive alternative to those used in previous work. The motivation for the choice here is to develop a low-cost alternative to alleviate cost as a limiting factor for the adoption of this technology.

The ARCS algorithm relies on spatial-temporal features extracted and a well-trained classifier. An analysis of the consistency of this algorithm has been conducted using the commonly used receiver-operator-characteristic curve. The benefit of data balancing as a critical step for improving the accuracy of the algorithm is presented. Data balancing is an

area of active research in computer vision and there is no universally acceptable solution. To this end, a strategy that incorporates two separate procedures is adopted, which proved optimal in this case.

Thesis outline

A review of commonly used visual monitoring techniques are described in Chapter 2, with emphasis on the application of computer vision in this area; although only a very limited number of work exists in this area. In addition, a literature review of the state of the art of target recognition methods is also given. Methods most applicable to the work are identified and rationalized.

The hardware design of a flexible and affordable system for 24hr visual monitoring is described in Chapter 3 using off the shelf consumer grade electronics. Several issues associated with this approach are given and calibration routines, designed to alleviate them are presented. In addition, a new online calibration routine for use at sea is proposed.

Two methods of estimating distance at sea are presented in Chapter 4. The first method deals with the case where the camera is uncalibrated and only diagonal field of view information obtained from the manufacturer is available. The second is based on camera pin-hole perspective model and expressions for estimating camera global motion from the horizon are given. A mathematical method for estimating the resolution of the system was also formulated given camera lens parameters; this enables the selection of the most appropriate hardware required for individual applications.

An algorithm for real-time automated localisation of the horizon in camera images is given in Chapter 5. The algorithm includes a pre-processing step based on the dark channel prior that can cope with the challenges caused by atmospheric conditions and sea state. In addition, it was shown that dark regions caused by swells and waves is often sufficient to dehaze an image in case of heavy fog. The wavelet horizon detection approach adopted also proves faster than the classical Hough transform method. The case of occlusion is dealt with by an efficient attitude filter based on extended Kalman filtering designed for fusing visual and inertial measurements.

A new algorithm for computer assisted detection of cetaceans at sea is proposed in Chapter 6. The algorithm includes morphological techniques that adapt to the changing image scene. This technique provides an advantage over other methods that rely on fixed threshold or fixed structuring elements. The algorithm is capable of coping with a considerable amount of noise in the challenging sea environments using Support Vector Machines (SVM) for classification. Steps towards training the SVM includes an effective data cleaning step based on Tomek Links.

The thesis concludes in Chapter 7 with a summary of the proposed algorithms and a discussion on future work is given.

Published papers

[1] Baruwa, A. L., Evans, A. N., Watson, R. J., & Wyatt, R., Video-based Real-time Automated Distance Estimation at Sea (RADES) for Marine Mammal Mitigation. Oceans 2013- Bergen, IEEE, June 2013.

[2] Baruwa, A. L., Evans, A. N. & Wyatt, R., The development of a remote sensing system with real-time automated horizon tracking for distance estimation at sea. SPIE remote sensing proceedings, September 2013.

In preparation for submission

[1] Baruwa, A. L., Evans, A. N., & Wyatt, R., Automated Recognition of Cetaceans at Sea (ARCS). IEEE Journal of Oceanic Engineering.

Chapter 2 : Cetacean Monitoring and Mitigation

This chapter reviews common methods of cetacean mitigation methods that are in operational use in the case of seismic operations and the application of computer vision in this area. A background to mitigation is provided in section 2.1 with a brief review of acoustic monitoring using PAM. In 2.2 a detailed description of the visual monitoring methodology used in marine mammal detection for real-time mitigation is given. A review of work done on improving visual monitoring of cetaceans using state-of-the-art camera technology in conjunction with computer vision is given. Finally, in section 2.3, we look at object detection techniques in literature that are most relevant to the problem here and a rational for the most relevant method is developed. Discussion and a summary of the chapter is given in section 2.4.

2.1 Background: Cetacean Mitigation in Seismic Surveys

Seismic surveys are undertaken by oil and gas industry to obtain sea bed maps with the aim of discovering mineral deposits. Surveys typically involve the use of an array of sound sources engineered to produce high amplitude sound impulses at regular interval of 10-15 seconds. The noise from the source is quite significant and can travel several kilometres underwater, hence the need for mitigation actions to protect marine mammals. There are various forms of mitigation that currently exist in practice and they can be grouped into three main categories:

1. **Temporal and Spatial Avoidance.**
   This is one of the most effective form of mitigation in practice since it involves restricting noise-pollution in times and/or places defined as sensitive to marine mammals e.g. migration routes, breeding area, breeding periods and feeding habitats etc. For example, the Marine Mammal Protection Zone in the Great Australian Bight is permanently closed due to the sensitivity of southern right whales and Australian fur seals [7]. In most regions, such areas classified as Marine Protected Areas (MPAs), often require special permits and more stringent mitigation procedures are in force [6]. Regulators often recommend planning surveys to avoid these places or times. Although this method is difficult to implement because of lack of sufficient scientific data [8], it is arguably the most effective form of mitigation.

2. **Operational Mitigation.**
   This includes single or series of operations that are carried out for mitigation purposes. Detection of marine mammal is not required for this kind of mitigation action to take place. These actions include:

a. ***Soft Start*:** This method involves gradual build-up of the air-gun sound levels with the aim of warning cetaceans and allowing them time to leave the survey vicinity before peak sound levels are reached.
b. ***Minimising Sound output***: This method of mitigation involves using air-gun array configurations that uses the lowest practicable volume [7] and minimizes sound propagation in at least one (e.g. horizontal) or all directions. Also, the use of alternative technology that can reduce sound levels is often recommended.
c. ***Shutdown***: In some regions, complete shutdown of the sound source is required in some certain situations e.g. at night when visual watch is impossible and other mitigation methods are unavailable.

3. **Detection Mitigation.**
   This is the kind of mitigation that occurs in real time alongside airgun operation. An area around the source is defined as the mitigation zone outside which, there is believed to be minimum impact on marine mammals. The most common safety radius is 500m around the source e.g. in the UK, USA and Canada and is mainly set to prevent physical damage to cetaceans [6]. When this kind of mitigation is in force, real-time monitoring of the mitigation zone to detect the presence of cetacean is required. Any detection of cetacean in this area would trigger mitigation action which usually involves complete shutdown of operation. Acoustic methods (mainly PAM) as well as visual monitoring by MMOs are the main methods of monitoring the exclusion zone. Aerial survey of the mitigation zone using special drones, helicopters etc. is another approach used in practice. However, this is not often used because it is relatively expensive and there is a lot of other logistics involved [6].
   Detection mitigation is an important form of mitigation because it allows seismic operation to carry on as per normal while still safeguarding marine mammals. It is essentially a win-win situation when done effectively. It is important to state at this junction that, the system developed in this thesis is one that enable real-time monitoring thus facilitation detection mitigation. In the following section, acoustic and visual monitoring activities currently used in detection mitigation are further explored.

2.1.1 Passive Acoustic Monitoring (PAM)

Acoustic monitoring can be either active or passive. Unlike passive acoustics, active acoustics involves capture and analysis of reflections of sounds emitted; it is not commonly used in marine mammal detection for mitigation. This is because, for obvious reasons, signals (especially low frequency ones) from active sonars could well cause some distress to the marine mammals they aim to protect. There is potential for the use of high-frequency sonars [7] but they are quickly attenuated and do not travel as far as low frequency sonars thus limiting detection range.

On the other hand, passive acoustics does not emit any sound but involves capturing sounds from the environment and analysing them. It can involve the use of mobile or fixed hydrophones [6]. In seismic survey mitigation, hydrophones are towed behind the vessel i.e. mobile hydrophones, thus, enabling the area round the noise source to be continually monitored. This also has the advantage of providing data in real-time for processing and near-real time [9] detection of marine mammals can be achieved.

PAM for mitigation purposes often involves the use of one or more linear array(s) of hydrophones towed behind the vessel with the aim of detecting and localizing marine mammal vocalizations. The simplest way of detections is the “manual approach” with specialists listening to sounds and/or looking at spectrograms [9]. The spectrograms, for example, can be calculated by taking fast Fourier transforms of blocks of incoming data [10]. Many automated detection methods have since been developed including analysing spectrograms e.g. to find intensity patterns [10], time series and wavelet analysis of signals [9]. Bearing and Range to detected cetacean vocalisation are often calculated using the time delay between hydrophones in the moving array. Please see [9] for more detailed overview. Figure 2.1 is an example of a fast Fourier transform spectrogram with automatic detection of Humpback whales.

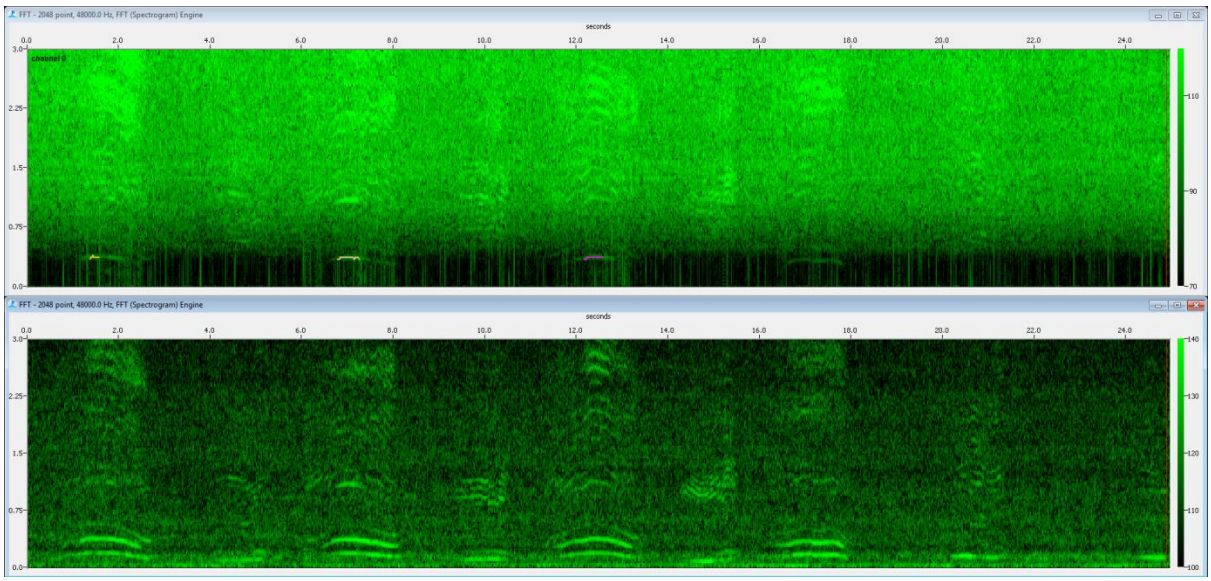


Figure 2.1: Spectrogram showing detection of Humpback whales using PAM [Courtesy of Seiche Measurement Limited]

Unlike visual monitoring (discussed in the next section), PAM can be used for 24-hour monitoring. It is often relied upon at night and in harsh weather condition when visual monitoring is impossible [6]. It is also useful for detecting mammals that are in the mitigation zone but are not visible. However, there are several factors that affect the effectiveness of PAM including:

1. It can only detect mammals when they are vocalizing. Some marine mammals are known to be much less vocal than the others especially Baleen whales.

2. The range of detection is dependent on background acoustic noise, for example, ship engine noise, air-gun noise and naturally occurring marine noise.
3. Marine mammal species with variable tonal sounds can be difficult to automatically detect and classify.

2.2 Visual Monitoring

Visual observation of the mitigation zone by MMOs is the traditional way of mitigation in Seismic operations and remains the most widely used. The unaided (human) eye is the primary visual monitoring tool for continuous scanning of the area surrounding the noise source, although, quite often, many MMOs often use a pair of binoculars. There exists a line of communication between MMOs and seismic crew which allows MMOs to advise them when a marine mammal is detected in the mitigation zone. The two main operations that make up visual monitoring of cetaceans are further explored below.

2.2.1 Detection of Cetaceans

Marine mammals will always come to the surface to breathe but how often they come to the surface varies from species to species. The traditional method of detecting cetaceans is through the human eye. Prominent features MMOs look out for are spouts or blows formed when cetaceans exhale, dorsal fins and tail flukes of the mammal itself. Breaching, splash from breaching and footprint of tail fluke are other ways of detecting cetaceans. Figure 2.2 and Figure 2.3 shows some of these common features. All of which, even from the biggest cetacean, can be difficult to detect because of several factors including reduction in visual resolution with distance. Also, observers can only focus on certain areas at a time. This is further exacerbated by the fact that mammals only surface for brief periods most of the time. Consequently, fatigue and experience of MMO generally affects marine mammal detection [6].

Figure 2.2: Common marine mammal features. Left: Southern right whale blow and Right: Southern right whale breaching.

MMOs are rotated regularly throughout the survey period (that can last up to several months) to reduce fatigue. Quite often, groups of two or three are employed at a time for

improved coverage and detection rates [8]. Binoculars are employed to improve resolution at distances and training is usually provided for inexperienced MMOs [6].

Visual Observation quickly becomes difficult and sometimes impossible for MMOs in adverse weather condition and as night approaches. Night-vision goggles have been employed for night time monitoring but this is considered ineffective due to reduced field of view [6] and its requirement for an illumination source on-board vessels [8]. A NATO funded research project [11] has designed and tested an infra-red (IR) binocular for marine mammal detection. Their results show that the system can be very effective, although it was affected by extreme weather conditions due to rough and hazy sea-states [11].

Figure 2.3: Left: Sperm whale fluke; Right: Pilot whales

Advances in computer vision techniques have facilitated the development of automated techniques for detection of cetaceans in thermal images. Podobna et al. have published a series of papers [12]–[14] describing a multispectral imaging system for marine mammal monitoring designed for airborne and ship based platforms. They developed a whale detection algorithm based on spectral and spatial processing. Results shown were only for aerial images and spatial processing was based on template matching.

Another automated whale detection system based on 360 degree IR video has been developed by Zitterbart et al [15]. They used a custom-built system including a stabilised gimbal, sterling cooled cryogenic sensors with 5Hz video stream of the entire field of view. Their algorithm uses support vector machine for classification and detection is done using short and long term averaging algorithm and dimensionality reduction through an Eigen-image algorithm.

Santhaseelan et al [16], [17] have also developed an automated whale blow detection system based on Neural Networks using IR video captured from shore. The focus was to help analyse videos captured to study the behaviour of animals. Their method is based on temporal and spatial characteristic of the blow and used a fixed threshold applied in a grid system to binarize the image. Further review of these algorithms is given in Chapter 6 and a review of computer vision algorithm used for target recognition is given in section 2.3. A good overview of automated methods designed to detect marine mammals is given in [18].

2.2.2 Distance Estimation at sea

Upon detecting a cetacean, MMOs must decide whether it is within the mitigation zone and, this requires an estimate of distance. Visual estimates by eye are currently used in practice but have been found to be inaccurate and biased [19] with the degree of error dependent on experience of observer and continued presence of target [20]. Continued presence of targets is likely to reduce errors in estimate but marine mammals only surface for brief periods at a time. While Baird et al [19] suggest that distance is underestimated, Williams et al [20] suggest that it is overestimated. In any case, wrong estimation of distance can either cause endangerment to cetaceans or force unnecessary shutdown of seismic operation which is very expensive.

Crude instruments such as sighting sticks (with pre-calibrated markings which correspond to certain angles when held at a certain distance from the eye) often used, are not very precise because, again, cetaceans surface for brief periods and vessel movement adds to the difficulty [21]. Reticule binoculars have been used for distance estimation by converting reticule values to angles [22]. This method is also not precise for similar reasons to the sighting sticks [19] but performs better than visual estimates by naked eye [22]. Gordon [21] reported that the use of laser range-finding binoculars capable of estimating distance to an accuracy of plus or minus 1m was not effective in estimating distance to cetacean at sea because target cetaceans were not big enough to be 'hit' by the laser and appeared only for brief periods.

Image ranging system at sea has since been developed by Hayashi et al [23] and a similar image based system has also been applied to estimating distance to cetaceans [21]. This method operates based on the same principle as the reticule binocular and uses basic trigonometric to calculate distance from the angle of dip measured between the horizon and target cetacean. This principle has been in existence for centuries [21], [23]. The image based method has been found to be the most accurate [20]–[22] but existing image ranging systems are not automated or capable of real-time operation making their usage in marine mammal mitigation virtually non-existent.

PAM is relatively new in the industry, but it is slowly gaining trust among users. But as mentioned previously, visual monitoring is the traditional method of marine mammal mitigation and arguably the most trusted method following the popular phrase "seeing is believing".

2.3 Computer vision for cetacean monitoring

Adopting latest technology to assist MMOs in detection of cetaceans can help improve the effectiveness of visual monitoring and enable visual monitoring during periods that would otherwise have been impossible. Technology can help improve range of detections, increase the number of detections and enable monitoring at night and in severe weather. However, only a few works in literature have been directed towards this, most of which are based on

feature extraction techniques applied to visual and infrared video signals. However, they demonstrate the benefits of technology in marine mammal mitigation. For example, detection ranges of up to 4km have been reported in [15] in the polar regions. Also, the multispectral camera system mentioned earlier, coupled with automated detection have been successfully trialled for marine mammal aerial survey and population estimate [13]; detection of marine mammals up to a couple of meters below water surface was reported.

Feature extraction techniques in computer vision may enable automated detection of cetaceans by mimicking the operation of MMOs to find features such as whale blows in video signals. Typically, it involves finding shapes and structures in the image signal that match the feature of interest.

The most basic form of feature extraction is thresholding. This is useful when illumination invariance is not required i.e. illumination of scene is likely to remain constant. Then object of interest of known brightness can be extracted by thresholding. There exist several strategies for thresholding [24] including; (1) global thresholding as in equation 2.1 where a constant threshold is used through an image (2) variable threshold where threshold value varies locally within the image depending on surrounding pixels and 3) dynamic thresholding where the threshold value varies due to spatial location of pixels.

$$g(x,y) = \begin{cases} 0, & f(x,y) < T \\ 1, & f(x,y \geq T \end{cases} \qquad 2.1$$

An alternative to thresholding might involve combining well known morphological operation of opening and closing with subtraction. Morphological opening is simply erosion of an image followed by dilation with a structuring element and closing is the dual operation which involves dilation followed by erosion. They can be used for identifying features based on their intensity by opening (or closing) an image and subtracting from the original image.

Thresholding and subtraction methods are not very robust since they are based only on illumination. In some cases a mathematical model that describes the feature is known i.e. a template is available and the shape is only flexible in terms of parameters that describe it [25]. In these cases, feature extraction is done by template matching where the image signal is searched for regions that correlates with the template or a known model based on predefined criteria.

2.3.1 Template matching techniques

As mentioned earlier, template matching has since been applied to cetacean detection based on aerial images of cetacean [13]. Given a template image, template matching can be done simply by centring the template image at a pixel location and counting the number of matching pixels; this is then repeated for the entire image. The position that best correlates with the template based on a predefined threshold is taken as the match. For a template *T*,

taking values in a window with coordinates *(x, y) ϵ W* and place at coordinates *(i, j)* of image *I*, a maximum likelihood solution to template matching is as given by the following equation [25]:

$$\min e = \sum_{(x,y)\epsilon W} (I_{x+i,y+j} - T_{x,y})^2 \qquad 2.2$$

By minimizing *e*, the position *(i, j)* in image *I*, that best matches the template can be estimated. Template matching is invariant to position and by relaxing the threshold value; it could be made relatively robust against noise but it is not invariant to size, orientation or illumination. This can be achieved by having multiple version of the same template with different orientation and sizes; as many as can be envisaged, although, this is clearly not efficient.

Template matching has a high computational cost of $O(N^2m^2)$ that increase proportionally with number of templates. This has led to development of techniques that less computationally complex based on Fourier transform. Scale invariance matching can be achieved with Mellin-transform while scale and position invariance can be achieved by combining Mellin with Fourier transform [25].

The Hough transform is another popular form of template matching for simple shapes (like lines, circles and ellipses) as well as arbitrary shapes (i.e. generalized transform). It is robust against noise and invariant to scale, rotation and size. However, it is very computationally costly and the mathematics involved may be rather cumbersome to accurately model complex shapes. In addition, in the case of generalized Hough transform, memory storage is needed to store look up table and accumulator and this can be substantial when object orientation and scale have to be considered [26].

The Hit-or-Miss Transform (HMT) is another template matching technique based mathematical morphology that can be applied to binary or grey images. HMT has the advantage of being generally faster since it is based on rank order operation and it is intensity invariant [27]. Traditional HMT was developed for feature extraction in binary images [28] using two structuring elements A and B called the foreground and background respectively. A is associated with the object or feature to be extracted and B is associated with the background. The HMT of a binary image *F* by *(A, B)* is the intersection of the erosion of *F* by *A* and complement of *F* by *B*:

$$F \circledast (A, B) = (F \ominus A) \cap (F^c \ominus B) \qquad 2.3$$

HMT has since been extended to grey scale in various papers [29]. The grey scale HMT (GHMT) of image signal *F* by greyscale template *T* can be obtained from an implementation proposed in [28] as:

$$F \circledast T = (F \ominus T) + (-F \ominus -T) \quad 2.4$$

where *F* is associated with the complement of the image signal and *T* represents the background structuring elements like in the binary method. Since erosion and dilation in greyscale images are expressed as minima and maxima 2.4 can be re-written for any location *k*:

$$(F \circledast T)(k) = \min_{n \in T}\{\big(F(n+k) - T(n)\big)\} - \max_{n \in T}\{(F(n+k) - T(n))\} \quad 2.5$$

The GHMT does not take a positive value and has a maximum of zero where *T* matches *F*. Another form of HMT for greyscale images proposed in [30] involves an operation called anti-dilation which is the dilation of image *F* by the negative of the template *T.* This version of HMT for greyscale is called Single Object Matching using Probing transform (SOMP) and it is simply the opposite of GHMT [30]. SOMP is obtained by subtracting the erosion of *F* from the anti-dilation by *T*:

$$F \circledast T = (F \oplus -T) - (F \ominus T) \quad 2.6$$

Several versions of the HMT for grey scale exist in an attempt to make it more robust to noise [27] [31]. The reader is referred to [29] for basic theory and detailed overview of existing grey scale HMT methods.

Of importance in any feature extraction techniques is the need for invariance properties in the technique adopted. In general, what is desired is invariance to illumination changes, noise, occlusion, position, size and orientation of target feature. While template matching is effective for aerial images it might not be directly employed for ship-board camera system. It might be useful in detecting certain features of a cetacean such as tail flukes such as Figure 2.3. However, the most common way of detecting marine mammal is by spotting their blows.

In cases where the shape of the feature of interest is not fixed and as such cannot be modelled, deformable shape analysis is required. Traditional techniques in these areas include deformable template matching, snakes, active shape models etc. [25]. The morphological approach is to use connected filters either on their own or combined with other image processing techniques and descriptors.

2.3.2 Deformable feature analysis

The most common way of detection cetacean used by MMOs is spotting their blows. In fact, it is possible to determine the specie of cetacean from its blow; for example, southern right whales blow have a unique “V”-shape. However, the shape of whale blow does not have a rigid shape but takes various shapes depending on, for example, camera view point, wind direction etc. as shown in Figure 2.4.

Figure 2.4: Blow of southern right whales; Left: the distinctive "V"-shape blow of a southern right whale; Right: "V" blow from a different view point.

Finding whale blows will require a feature extraction technique that is invariant to shape, size, orientation, position and illumination changes. The traditional approach to matching features that vary in shape is deformable template matching [32]. The idea is that a template can be deformed based on pre-defined constraints to align with similar shapes. A popular variant of the deformable template technique was introduced by Yuille [33] for face recognition. The approach was to fit a hand-crafted model of eyes using a gradient decent optimization technique [32]. The deformable template approach is an energy maximization process [25] whereby parameters that define a shapes deformation contribute energy (e.g. edge energy) and the combination of these parameters that maximises energy are sought after.

An active contour (or snake) is another way of extracting deformable features. In a general sense, the technique is a process of evolving a set of points (i.e. the contour) to match image data [34] rather than evolving a shape in the previous approach. Active contour describes a feature by enclosing, this is achieved by allowing the initial contour to shrink based on predefined constraints. Snakes have the advantage over deformable templates that they are defined by fewer parameters [25]; however, it involves an iterative approach which can be slow. Also, result achieved depends on the contour initialisation and contour which are difficult to choose automatically [34].

The more recent approach to deformable feature extraction is Active Shape and Active Appearance Models (ASM and AAM respective). In the ASM, a flexible shape is modelled to be based on several landmark points. Traditionally, landmarks are manually chosen [35] in each training image, with each landmark representing the same location on object. Once landmarks are selected, corresponding landmark in all training images are aligned. This is achieved by rotating and translating consecutive shapes until they are aligned [36]. The variation in these points is called point distribution model. Principle Component Analysis (PCA) is then applied to capture the statistics of the variation in the point distribution model [34].

PCA defines by how much a shape can change (based on the training data). In ASM, feature is extracted in an iterative process to match the objects point distribution model to points in

the image. Just like snake, this technique allows the model to deform based on the statistics captured in the principal component analysis to match image data. It has been applied a lot for face recognition and image retrieval with good results; however, it is computationally costly. Also, the modelling of shapes in terms of landmark points and consequently, the manual selection of these points makes it impractical to use. Although, recently, automatic landmark selection techniques have been developed e.g. in [37] and [38]. Finally, the effectiveness of ASM is dependent on the number of training data and, just like snakes, poor initialisation can affect its performance [33].

Another framework for flexible shape matching using a descriptor termed shape context has been developed in [32]. Shape context describes the distribution of the rest of the shape with respect to a given point on the shape [32]. Like ASM, it uses discrete points on an object to model its shape but unlike ASM, these are random points rather than landmarks. Feature extraction here is as simple as finding the corresponding point in an image that matches (or has maximum similarity) to the shape context of the model shape. This technique was reported to be invariant to scaling and translation and robust to occlusion and presence of outliers [32]. It can also be made rotation invariant by modifying the computation of the shape context.

Finally, the morphological alternative for flexible/deformable feature extraction is perhaps, attribute morphology. Attribute morphology consists of a class of connected set opening and closing operation first introduced in [39]. Unlike classical morphology operators, attribute morphology does not use a fixed structuring element. Like previously described deformable feature extraction techniques, it adapts to fit the content of the image based on a predefined stopping criterion. Area morphology is a class of attribute morphology whose stopping criterion is area size. Other attributes/descriptors can be used as stopping criteria including contrast, compactness, volume, energy or power etc. [40]. Attribute openings and closings are regional operators that lower the regional maximal or raise the minimum respectively. This technique requires no template or model, just a suitable descriptor as stopping criteria. Efficient algorithms have been developed and a good comparison exists in [41].

As mentioned previously, computer vision for marine mammal monitoring is a new area of research and only a handful of works have been directed in this area. It is clear from the previous section that template matching techniques cannot be suitable for extracting features that vary a lot from image to image like whale blows. However, this is arguably, the most reliable feature for detecting whale and has been used for centuries. Thresholding and subtracting, used in some previous work [16], are not ideal in environment with constantly changing illumination. Realistically, deformable feature extraction techniques offer the best chance of success.

Snakes and Active Shape Model (ASM) are iterative and require initialisation so they may not be efficient for real-time application. Deformable template still requires a template that

can be deformed and this may be difficult to model since whale blows are completely without form and the shape can be very dissimilar from image to image. Therefore, the conclusion is that, the most promising techniques are those based on descriptors.

Finding the optimal descriptor or set of descriptors that are partially or completely invariant to shape can facilitate a robust real-time detection system. The techniques reviewed previously that fit this criteria are shape context [32] and attribute morphology [39]. More detail about work to be done in this area is detailed in Chapter 6.

2.3.3 Classifiers

Once a feature has been extracted, a set of descriptors that define the region in the image may be computed and passed to a classifier to determine if the region is a feature of interest i.e. a whale blow or not. Some of the most commonly used classifiers include K-means clustering, Artificial Neural Networks (ANN), Support Vector Machines (SVM), Deep Neural Network (DNN) and a host of other machine learning algorithms. As mentioned before, artificial intelligence method based on neural method has been developed for detection and classification of whale blows [15][16] as a preference to rigid template matching techniques.

Classifiers typically involve building a mathematical model that generalizes the given problem using a set of examples known as the training data so that its ability to classify unseen data is as 'optimal' as possible. There are two main types of classifiers, supervised and unsupervised. A supervised classifier is one where all the training samples are categorized into classes (e.g. blow event or no events) while unsupervised classifiers try to find the appropriate categories for each sample. These two classifications are extremes and there several other classifiers in between.

Although an unsupervised classifier can be effective in certain problems and capable of modelling complex problems, supervised classifiers are arguably the simplest, more intuitive to implement and thus, the only classifiers explored here. Also, only discrete classifiers are reviewed here since the aim is to label events simply as blow events or not. Discrete classifiers aim to map continuous variables or inputs into simple discrete output.

Given a fully labelled set of training samples, each belonging to one of n classes $w_1, w_2 \ldots w_n$, one approach of classifying new sample is to find the class w_c to which it is closely related to by computing its nearest neighbours; this is a technique known as K-nearest neighbours. The nearest samples may be computed using distance measures such as Euclidean, Manhattan or Bhattacharyya distances. The class to which the sample belongs to is then inferred from the k samples e.g. by majority voting etc. A major drawback of this method is determining the appropriate value of k and deciding which distance measure to use. Also, simple distance techniques (like those already mentioned) are linear and thus may not be effective in problems with non-linear relationships.

Machine learning classifiers are increasingly becoming popular due to their ability to generalize complex non-linear models. ANN is one of such algorithms that operates by trying to mimic the human brain (neural network). ANN seeks to map input to output using a set of nodes (similar to neurons in human) and weights (similar to synapses) that connect nodes from one layer to another. There are numerous types of ANN but the one described here is called the multilayer perceptron.

Support Vector Machines (SVM) are an attractive alternative to ANN that operate by using mathematical modelling to find a hyperplane that separates two classes. They are particularly well suited for binary classification i.e. labelling events simple as blow or no blow. In practice, this involves transforming the training samples to a higher dimensional space using a function called a kernel and then finding the hyperplane in that space. SVM is considered much easier to use than ANN [42] since the latter requires careful selection of the number of nodes and layers that make up the network.

2.3.4 Multi-sensor systems

Multi-sensor systems for cetacean detection have previously been investigated in [12]–[14]. The use of complementary sensors can quickly widen the scope or range of information available for an application and as a result improve the robustness of the system. For example, the use of multi-spectral camera system can; 1) improve monitoring in complete darkness (if a thermal camera is available), 2) enable detection of marine mammal a few meters under the water surface [12]–[14] etc.

2.3.4.1 Multi-sensor data synchronisation and alignment

As consumer electronics are increasingly becoming cheaper, it is becoming increasing common for engineers and researchers to combine multiple sensing systems in their applications. However, one of the main challenge facing this type of system, other than actual fusion of the collected data, is synchronisation. Using measurements from sensors that are taken at separate times can severely undermine the precision, certainty and overall performance of a system. Arguably, the easiest way to synchronise data from multiple sensor is to put them on the same reference clock and trigger system such that each sensor collects a measurement sample at the same instant in time. In this kind of system, the frequency of data acquisition will be constrained by the sensor with the longest acquisition time [43]. In addition, typically, systems like this must be custom built from scratch thus making it rigid and not easily scalable.

In multi-sensory applications, it is becoming increasingly common to use commercially available sensing units that are wholly separable [44] with each unit having a separate clock. The lack of an application wide reference clock means that there will be lack of perfect timing information i.e. resulting in an asynchronous system. This setup is highly flexible and scales easily. Often, it is also cheaper and arguably takes less time to build. The main

challenge here is the non-deterministic nature of sample instant from each sensor, making data synchronisation and fusion difficult. As a result, this field is attracting a lot of attention and quite a number of works have been directed towards solving this problem [44]–[47].

The problem of synchronising multiple sensors can be divided into three including 1) estimating the time a sample measurement from each sensor was taken 2) time synchronisation and temporal ordering of multi-sensor data and 3) spatial synchronisation of multi-sensors data.

Estimating sample timestamp has been considered in [44] for real-time implementation based on Kalman filtering of arrival timestamp of samples from each sensor on the processing platform (e.g. a PC). The use of a Kalman filter enables the tracking of the individual clock drift of each sensor and estimate sample intervals of the sensor. Their work shows the need for filtering and this is supported by [45] based on results from comparing synchronisation performance with and without filtering. Here [45], a simple calibration rig for offline inertial measurement and camera sensors time synchronisation was introduced. The result of the synchronisation is an estimation of the constant delay between the two sensors. Their work combined the time estimate technique in [44] and a temporal alignment algorithm.

Temporal alignment or synchronisation simply involves estimating the measurement delay between sensors. Two temporal alignment algorithms based on cross-correlation and phase congruency have been compared in [46], also for synchronising inertial measurement and camera sensors. Both techniques are based on analysing a sequence of rotation estimates measured from both sensors. The cross-correlation approach involves finding the time delay that maximizes the correlation of measurement sequence from both sensors. While the phase congruency method involves analysing the measurement sequence in frequency domain to establish the phase difference and hence time delay.

The previously described methods of time synchronization are carried out independently, without any knowledge of the spatial alignment of the sensors. The problem of spatial alignment involves estimating the relative orientation of a sensor from another. Work in [47] and [46] include spatial alignment and joint alignment in space and time using extended Kalman filter has also been considered in [48].

It is important to stress here that, choosing between synchronous and asynchronous sensor configuration is application specific. Synchronisation of asynchronous sensors system is well studies and several techniques have been reviewed. Once the problem of temporal and spatial synchronisation is solved, fusion can be done using popular techniques such as Kalman filter [44]. A review of fusion algorithms and techniques is described next.

2.3.4.2 Multi-sensor Data fusion

Data fusion as described here refers to the scenario where information is retrieved individually from each sensor and then combined to provide an optimal solution to a problem. We assume that raw data from various sensors have been acquired and processed and we seek to merge the noisy measurements extracted. This differs, for example, from techniques that fuses raw data from sensors.

Based on the amount of literature found using these methods, we conclude that the most common method of using noisy measurements in computer vision and robotics are probabilistic methods based on state estimation. These methods are especially attractive in dynamic system; they are commonly known as tracking techniques and they aim to determine the state (e.g. position) of a system given a set of observation or measurements [49]. Other methods are based on fuzzy set theory, rough set theory and random set theory [50]. Here, we focus on the probabilistic methods because they are widely applicable in situations where we are dealing with data uncertainty [50].

State estimation systems approach the problem by constructing Probability Density Function (PDF) to characterise data uncertainty. When the PDF is Gaussian, and the system model is linear, the most optimal solution is known as Kalman filter. In practice, the system is often non-linear and the PDF is not Gaussian. The Extended Kalman filter (EKF) provides a technique for dealing with this using a Jacobian to linearize the system. Another variant known as the Unscented Kalman filter (UKF) is becoming increasing popular and it does not have a linearization step [49].

A particle filters is another method that rely on PDF, it approximates the PDF using samples called particles. As a result, it can represent non-Gaussian PDFs as well as non-linear state models. It is conceptually more robust than the Kalman filter but more computationally costly. Maximum Posterior techniques assume that the PDF is known a priori either analytically or empirically; thus given a series of observations, the optimal state is one that maximizes the probability distribution [49].

It is important to state here that these techniques are implementations of a broader class of filters known as Bayesian Filters. The Kalman filter and Extended Kalman filter are the most widely and the preferred technique here because they are robust and theoretically well understood.

2.4 Summary

In this chapter, various methods of mitigation employed in the Seismic industry were discussed. Detection mitigation is arguably the preferred type of mitigation since it allows simultaneous protection of marine mammals and seismic operation. This method is often used in conjunction with one or more operational mitigation methods to better protect marine mammals.

While PAM relies on state of the art technology for detecting and localising of cetaceans, visual monitoring relies principally on humans for visual detection and estimation of distance at sea. Only a handful of works has been directed towards computer-assisted method for visual monitoring. The review of existing technology shows that there is potential to improve visual monitoring, thus, the remainder of this thesis details the development of two major components of a computer-assisted system that have been identified as having the most potential to improve the visual monitoring: 1) a tool to provide accurate distance information to visual observers; 2) a tool to improve detection of cetaceans.

Rigid and deformable computer vision methods that exist in the literature have also been reviewed with deformable methods considered to be the most relevant in our application. Also, it has been shown that, the use of multiple (complementing) sensors can quickly widen the scope of information available for this application, in the next chapter, a description of hardware components that make up the multi-sensor system is given. In addition, sensor alignment techniques are implemented following findings from the review done here.

Chapter 3 : Camera monitoring system (CMS) design

The Camera Monitoring System (CMS) can be considered as a subset of the RHVM system that is responsible for the collection of data from sensors and delivering this data to the software algorithms that process them. In addition, CMS provides a method for sending the result of the processed data to a remote location where they are displayed and monitored i.e. the monitoring station. In the case of a Seismic operation, this system will be typically located on a vessel. An RHVM system consist of CMS (mainly hardware) and a series of software algorithms described in later chapters.

CMS is made up of individual commercially available sensors which accelerated the hardware design, since development of bespoke solutions can be very time consuming. It also allows flexibility in the modern world where the rate of technological development is fast moving. The consequence of this approach is a multi-sensing system whereby output from the sensors are completely unaligned both in space and time. Various calibration routines required to deal with this are rationalised and duly introduced. In addition, the method for estimating camera parameters using a planar pattern is also presented; this is required to understand 3D features captured in a 2D image.

A major contribution of this chapter is the presentation of a new online calibration technique that does not require any calibration pattern. This technique relies on the detection of the horizon line using the horizon tracking system presented in Chapter 5. It is suitable for online camera calibration as well as estimating the misalignments between sensors. The approach has been tested extensively using real and simulated data.

In Section 3.1, the hardware components that make up the CMS are introduced. In Section 3.2 the methodologies used for calibrating the primary camera sensors are described. Section 3.3 describes the various calibration routines required for reliable fusion of information gathered from the multi-sensor front-end. A new technique for online calibration is introduced in Section 3.4 and a summary of the chapter is given in Section 3.4.

3.1 System components for the camera system

CMS in its most basic form consists of a single camera at the front end and a computer running the computer vision software (described in the next chapters). The software consists of a C++ program with a user friendly Graphical User Interface (GUI) designed for easy operation. It grabs images from the cameras, processes it and displays the result on a screen.

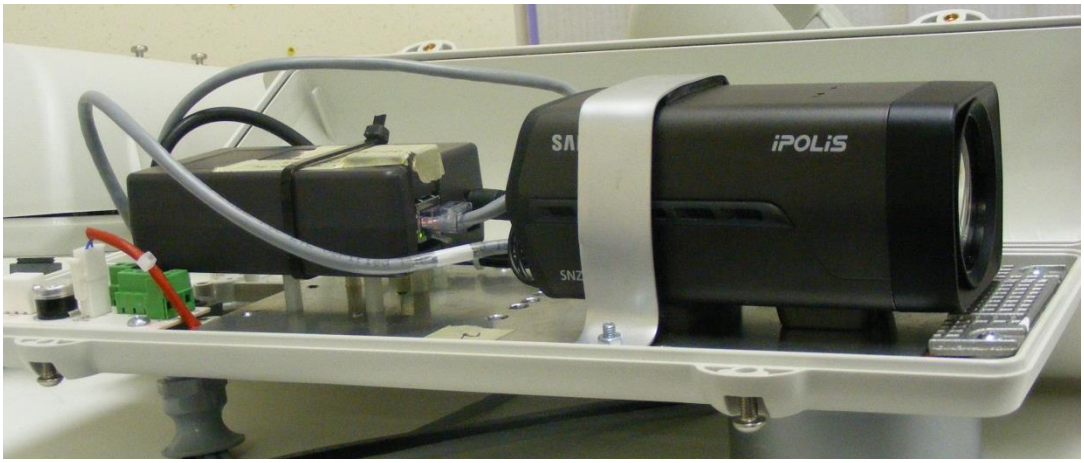


Figure 3.1: Single camera system

3.1.1 Multi-sensor System

The single camera system typically consists of a High-Definition (HD) visual camera and the processing software. However, this is limited to daylight operation only and has a limited Field of View (FOV). A multi-sensor system was developed to include a long wave infrared camera to enable 24-hour operation and Pan and Tilt Unit (PTU) to extend the FOV as shown in Figure 3.2.

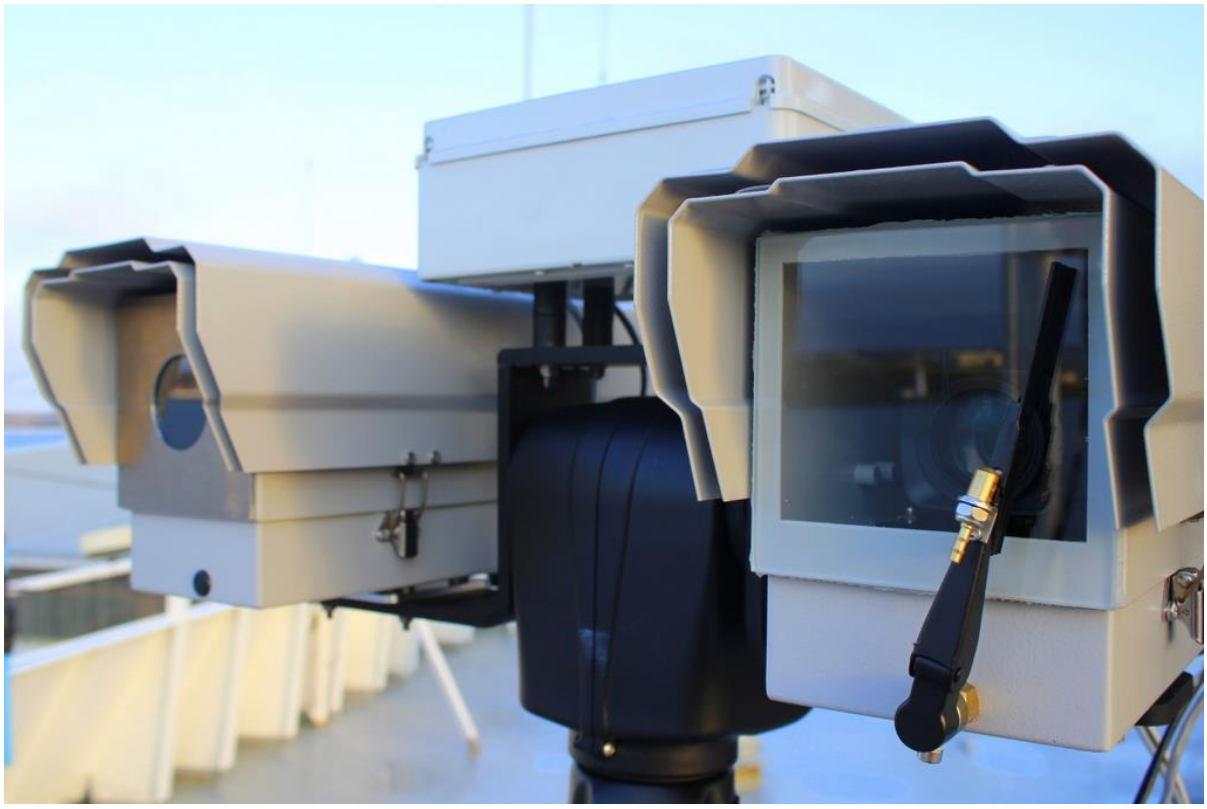


Figure 3.2: Multi-spectral camera system

This system facilitates coverage of a relatively large area on the sea surface using the panning system with the limitation that only a section of the area is visible at any given time. However, this is not likely to be a problem since experience suggests that it is unlikely

to have more than one area of interest at any one given time. In fact, this technique is in-line with current MMO methods whereby sections of the sea are scanned. Whereas, for humans, repetitive scanning is a mundane task that can, very quickly, lead to fatigue, PTUs do not suffer such. It is important to stress at this point that the aim is to find the system that is most effective and affordable.

Other optional sensors integrated here include a GPS and Inertial Measuring Unit (IMU). The IMU typically has nine (9) Degrees of Freedom (DOF) using a three-axis accelerometer, gyroscope and magnetometer to provide real time attitude information e.g. yaw, pitch and roll angles. A microcontroller is used to interface the IMU and GPS with the processing computer. All sensors used in the system are Ethernet based, allowing the processing computer to be in a suitably remote location as show in Figure 3.3. This flexibility is key to achieving monitoring from remote location as described in the next section.

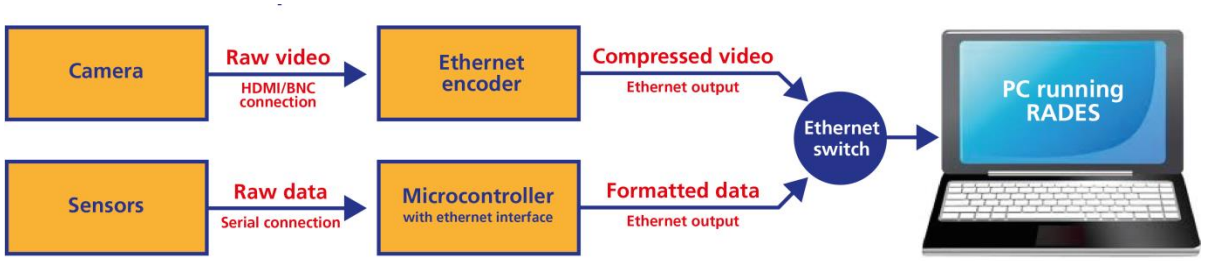


Figure 3.3: Block diagram of CMS

3.1.2 Monitoring station

The monitoring station is the location where data gathered and processed by the software system are displayed to a human observer. It is made up of a few screens where images augmented with graphics generated by the processing (software) system are displayed, evidence of detections is reviewed and captured by operators and decisions are made. Whereas the communication link between the front-end system and the processing system is a fixed wired Ethernet connection, the link between the processing system and the monitoring station could be wired, wireless or both depending on where the latter is located relative to the offshore platform. The options are:

1. A fixed wired connection when location on the same platform
2. A direct wireless link when located on another offshore platform or onshore location with line of sight to observation platform.
3. A satellite link when located on another offshore platform or onshore location without line of sight to observation platform as show in Figure 3.4.

3.1.2.1 Remote monitoring

Traditionally, MMOs are required to monitor the area around the vessel and they do this on the deck of the vessel where they are exposed to hazards such as wind, rain, cold as well as

other health and safety risks. The RHVM systems alleviates these risks even when the monitoring station is on the same vessel since it is co-located in a secure area below deck.

When the remote monitoring station is located onshore, HSE risk is further reduced since the process of transporting observers on and off the vessel offshore will be eliminated. Since the processing system is always located on the same vessel (platform) as the sensing unit, no information is lost during processing. But the limited bandwidth typically available in wireless system provides a major challenge for reliable communication even with commonly used data transfer/sharing methodologies.

During initial trials, VNC was used to achieve satellite remote monitoring (as described in section 4.3). However, the subjective quality obtainable using this setup, at relatively higher frame rate, was quite poor. As a result, frame rate was sacrificed for quality, and it was found that 0.3fps is the average achieved with reasonable subjective quality.

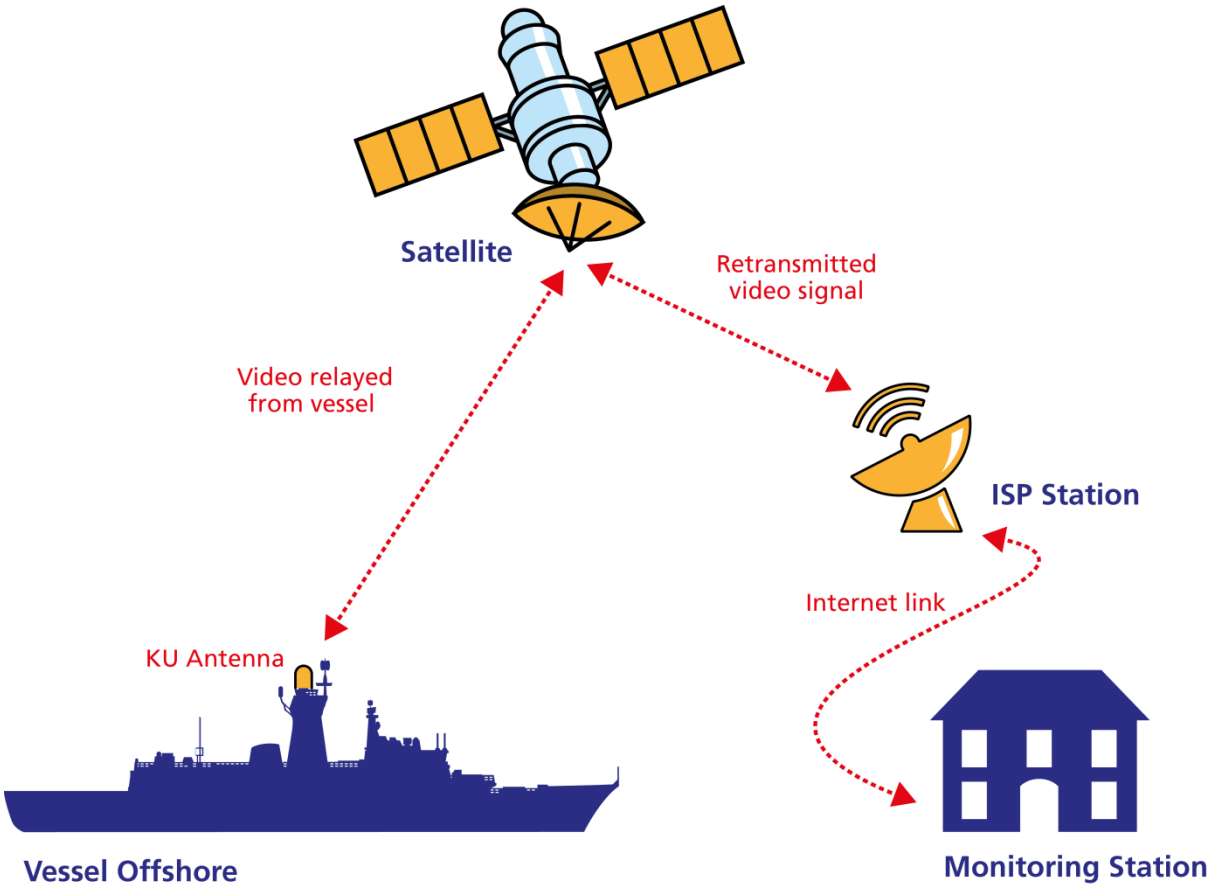


Figure 3.4: Graphical representation of RHVM system with remote monitoring from a location onshore

The reason for the poor frame rate is because most remote display protocols are optimized for very low-frequency update applications [51]. While the ability to control the remote computer in TightVNC is highly desirable, the poor performance in terms of video data transfer is not. To achieve a better frame rate and video quality, larger bandwidth is required (which is expensive) and even more so is a better, more efficient compression algorithm.

MPEG-2 and H.264 video codecs are more bandwidth efficient for high-frequency update application since they both employ motion estimation for inter-frame coding. For example, authors in [51] have developed a hybrid remote display system that relays graphical data to the end user through traditional VNC/RBF protocol or through Video streaming in H.264 format. They employed a decision heuristic based on motion estimation to determine which method is used to relay images to the client. Here, the strategy employed is to use two separate communications channels, one for control and the other for data transfer.

Software encoding and streaming images with accompanying metadata has been implemented in the RHVM system. Each image from the camera is sent to the remote station with additional metadata including 1) localised horizon (i.e. for vessel orientation determination and mitigation zone demarcation) 2) pixel coordinates of detected cetacean and 3) additional sensors data from GPS and IMU. All cameras have an individual stream each and H.264 is used for video encoding while metadata are sent as base64 encoded binary data. The offshore processing software is controlled on a separate communication channel by plain-text commands sent over a secure link.

Preliminary test shows that this approach facilitates better remote monitoring by improving the quality of data transfer over the communication channel. It offers flexibility in terms of control of the data-rate as a trade-off between frame-rate or picture quality as required.

3.2 Camera Calibration

The camera needs to be calibrated to obtain its intrinsic properties before it can be used for any reliable operation such as distance estimation. The pinhole camera model is adopted here.

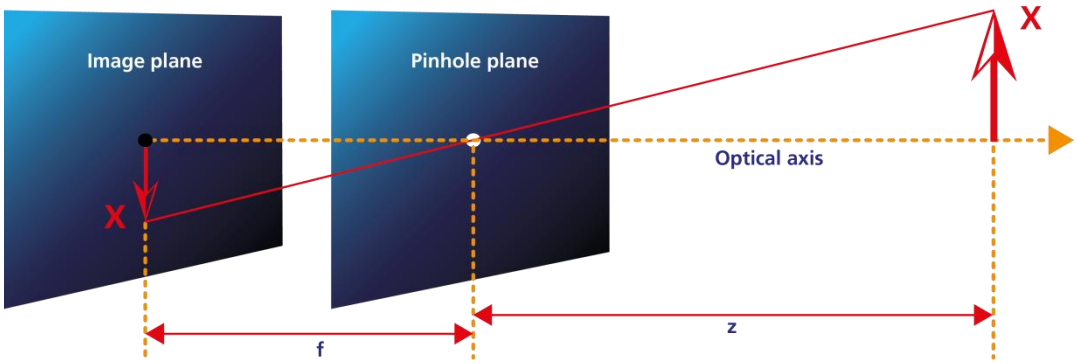


Figure 3.5: Pinhole camera model

The model is represented as having a very small aperture at its centre of projection as shown in Figure 3.5. Light from the scene enters the camera through the hole and projects an image on the image plane behind the pinhole. For an ideal pinhole camera, the distance between the hole and the image plane is the focal length. Unlike the pinhole, modern day cameras have a lens at their centre of projection that is used to focus light from a relatively

larger aperture on the image plane [52]. The distance between the lens and the image plane is the focal length of the camera.

3.2.1 Single Camera Calibration

Two methods of calibration were developed and evaluated in this thesis as described next.

3.2.1.1 Pin-hole method

Based on similar triangles in the pinhole model as can be seen in Figure 3.5, a point at *X* in a scene is related to its projection on the image plane *x* based on the following expression:

$$\frac{x}{f} = \frac{X}{Z} \tag{3.1}$$

Z is the distance of the point X from the hole and *f* is the focal length. This means that if the picture of a target of known size is taken at a known distance from the lens of the camera, it is possible to calculate the focal length of the camera. If the size of the image plane (i.e. camera sensor) is known, the size of the target *(x),* in mm, on the image plane can be calculated from:

$$x = x_{pixel} \times \frac{D_{mm}}{D_{pixel}} \tag{3.2}$$

x_{pixel} is the size of the target on the image plane in pixel while D_{mm} and D_{pixel} are the sizes of the image plane respectively in mm and pixels. Figure 3.6 shows a target consisting of a triangle of red circles 25.4mm apart from each other. A simple algorithm based on region growing was developed to automatically look for two targets of red colour and calculate the pixel distance between them.

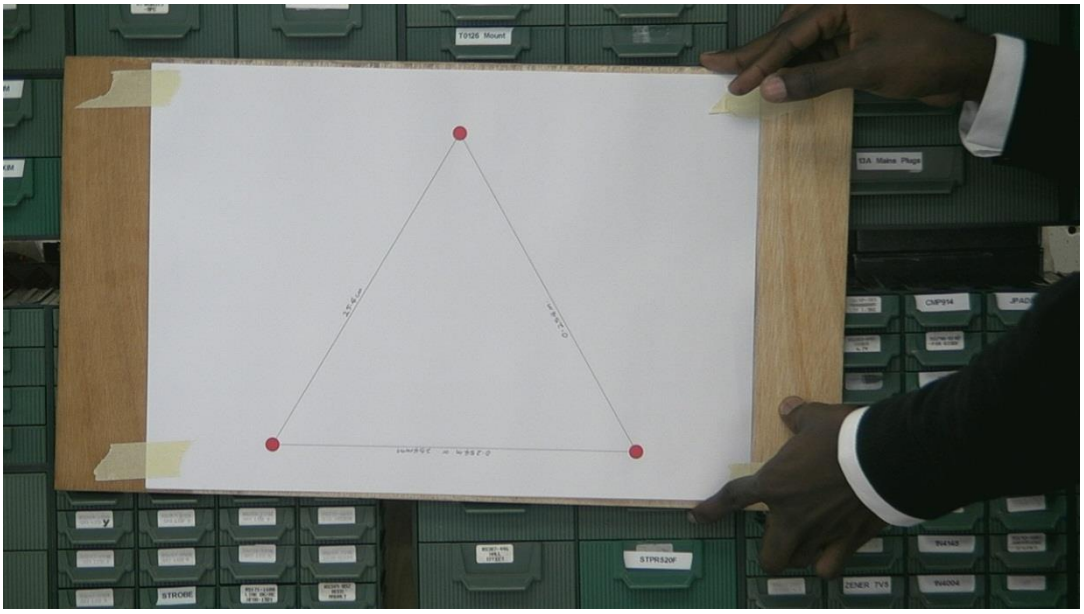


Figure 3.6: Triangle of targets at equal distances from each other

The algorithm operates by first searching the image for the pixel coordinate that is closest to the colour red. This is done by finding the minimum of the distances between the vector of each pixel in the image and the vector of the colour red. This pixel is then taken as the seed which is grown by grouping neighbouring pixels that meet a predefined criterion. The centre of mass of the region grown is taken as the centre coordinates *(xc_1, yc_1)* of the target circle. This process is repeated to find the centre coordinate *(xc_2, yc_2)* of the second red circle. Then the pixel size of the target is given as:

$$x_{pixel} = \sqrt{(xc_1 - xc_2)^2 + (yc_1 - yc_2)^2} \quad 3.3$$

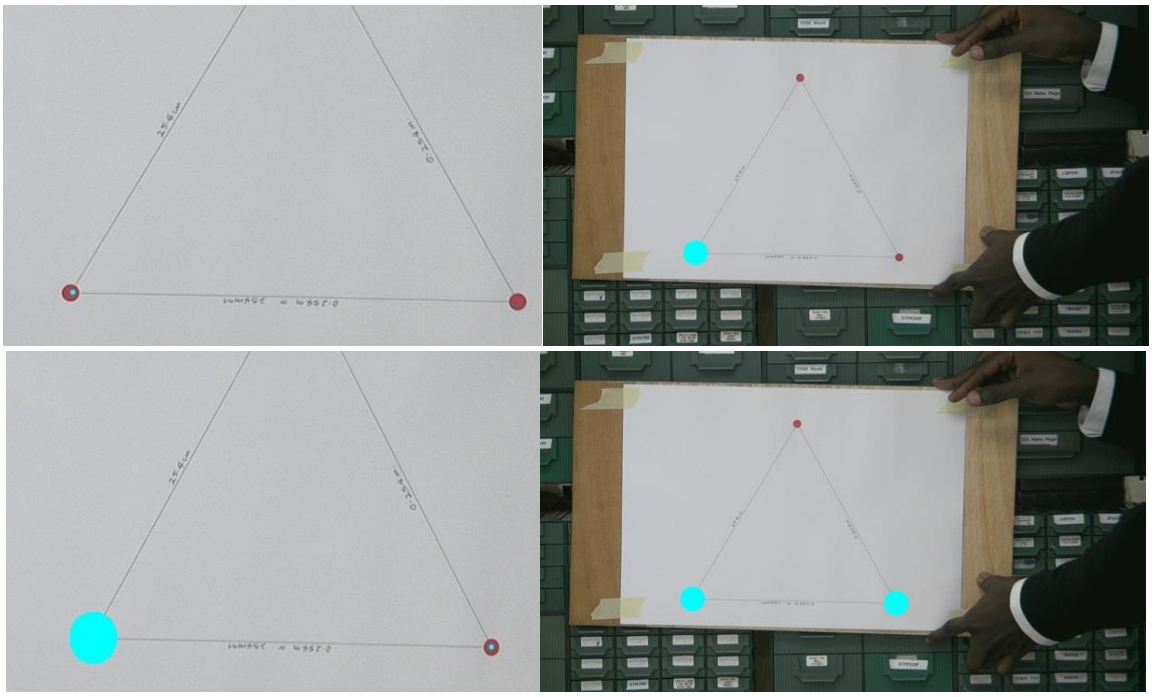


Figure 3.7: Result of red target detection system; Top: detection of first target; Bottom; detection of second target.

Figure 3.7 shows result of the target finding algorithm based on region growing. The target is placed a known distance from the camera where its image is in focus. From the equation 3.1 and 3.2, the focal length in mm is given as:

$$f = x_{pixel} \times \frac{D_{mm}}{D_{pixel}} \times \frac{Z_{mm}}{X_{mm}} \quad 3.4$$

since,

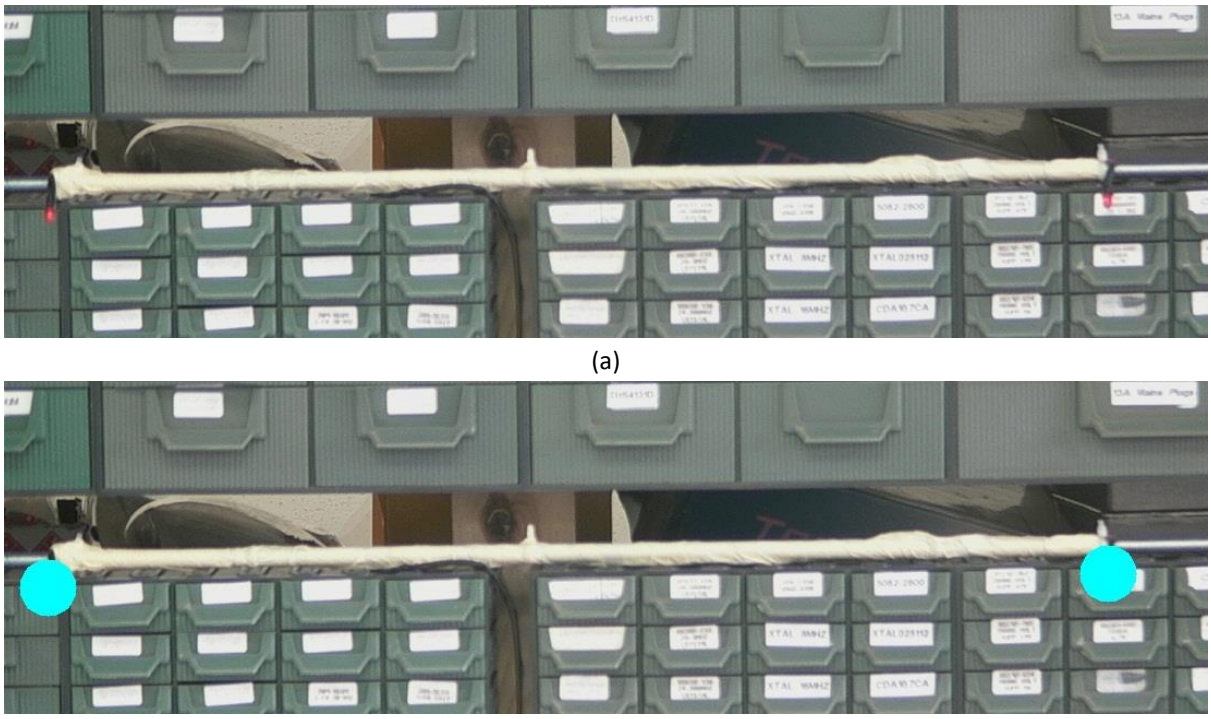
$$FOV = 2\ \tan^{-1}\left({}^{D_{mm}}/_{2f_{mm}}\right) \quad 3.5$$

This can be simplified to:

$$FOV = 2\ \tan^{-1}\left(\frac{D_{pixel} \times X_{mm}}{2 \times x_{pixel} \times Z_{mm}}\right) \quad 3.6$$

The FOV of a camera can be quickly found using red targets system and equation 3.6. The reason for choosing the colour red is because it is prominent on a white background. Also,

the algorithm has been tested with two red LED with known separation. It is assumed that red LEDs would be conspicuous in sea or sky background, in which case, this method can be useful for online camera calibration in future work.



(a)

(b)

Figure 3.8: Result of red LED automatically detected by the red target detection algorithm. (a) Original image; (b) result of detection.

3.2.1.2 Multiple poses of a planar pattern

The quick method does not consider, the introduction of the lens in the camera model and errors inherent in the camera construction process. The use of a lens introduces distortion in the image which can result in error in the estimate of the pixel size of target using the method described previously. A modified version of the Pin-hole model is more commonly adopted where the pixel coordinates *(x, y)* of a point P, whose coordinates relative to the camera centre of projection is (X, Y, Z) are related by:

$$x = f_x \frac{X}{Z} + c_x \;\; ; \qquad y = f_y \frac{Y}{Z} + c_y \tag{3.7}$$

where c_x and c_y represent the displacement of the image centre coordinates of the sensor from the optical axis in the *x* and *y* direction respectively [52]. Equation 3.7 can be expressed in matrix form by defining a parameter M called the camera matrix such that:

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

therefore,

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_C \quad 3.9$$

Dividing through by *w* the relationship in equation 3.7 can be recovered from the matrix form in equation 3.9.

A camera calibration method based on multiple poses of a planar object was developed based on the model in equation 3.9 [52], [53]. This method is adopted as a more accurate alternative to the pin-hole method. The algorithm uses a homographic approach (i.e. projection mapping from one plane to another) to find the intrinsic and extrinsic camera parameter. The intrinsic parameters of the camera include the camera matrix *M* while the extrinsic parameters consist of a rotation and translation matrix that maps a real-world point from global coordinate space to the camera coordinates i.e.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_C = R\left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_G + T\right) \quad 3.10$$

where *R* is a 3 by 3 matrix and *T* is a 3 by 1 matrix, subscripts *C* and *G* refers to the point in the camera and global coordinates respectively. If *P* represents a point in the global coordinate and *p* is the same point on the image plane equation 3.10 becomes *p = MR(P+T)* which is solved mathematically by locating *k* points on a planar object of known position in the global coordinate in multiple poses (the reader is referred to [52] for more background information).

The camera calibration algorithm uses a “circle grid” pattern as its planar object since it is well suited for calibrating the thermal camera as well as the HD camera. The pattern is made of aluminium sheet with holes of fixed size and equal spacing punched through. The perforated aluminium sheet is then placed in front of a black heat source. The aluminium sheet masks some of the heat radiated from the heat source forming the desired pattern on the thermal imager. This technique follows the strategy developed in [54]. The heat source is chosen as black so that a similar light on dark pattern is visible in the HD camera.

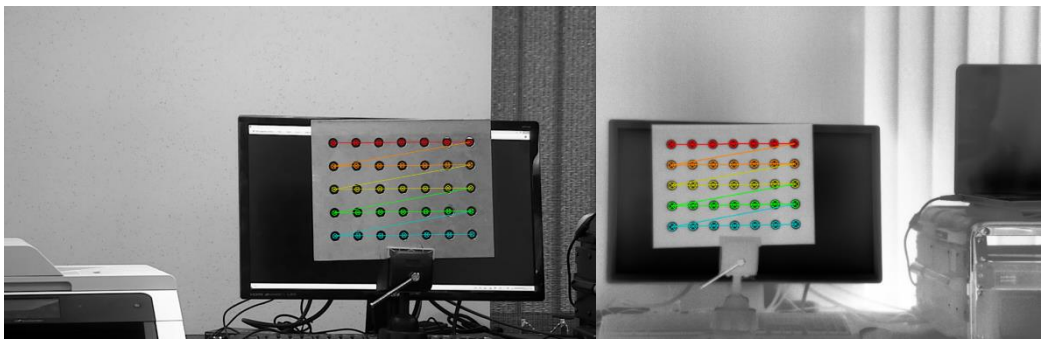


Figure 3.9: Circle grid pattern detected by the calibration utility in visual camera (left) and thermal camera (right)

Figure 3.9 shows example of circles found using the calibration algorithm. The spacing between the circle patterns are known and used to develop mathematical equations to solve for *M*. Multiple poses of the pattern result in multiple equations that is then used to reduce error and accurately estimate the camera matrix [52]. Recall that the FOV of the camera can be calculated as:

$$FOV = 2\ \tan^{-1} {}^{D_{mm}}/_{2f_{mm}} \tag{3.11}$$

where D_{mm} is the real diameter of the image plane and f_{mm} is the focal length both in millimetres (mm). If W_{mm} is the width of the image plane in mm and '*a*' is the ratio the real-world height to the width of the image plane, then:

$$D_{mm} = W_{mm}(1 + a^2)^{1/2} \tag{3.12}$$

since,

$$f_{mm} = f_x \times s_x \tag{3.13}$$

and

$$s_x = {}^{W_{mm}}/_{W_{pixel}} \tag{3.14}$$

we have,

$$FOV = 2\ \tan^{-1}\left(\frac{W_{pixel}(1 + a^2)^{1/2}}{2f_x}\right) \tag{3.15}$$

where, W_{pixel} is the pixel width of the image plane and s_x is the size of a single pixel in the horizontal direction. The parameter '*a*' can also be obtained from:

$$a = \frac{f_y}{f_x} \times \frac{W_{pixel}}{H_{pixel}} \tag{3.16}$$

Using the parameter of the camera matrix, the FOV of a camera can be found using equation 3.15. In addition to solving the intrinsic parameters of the camera, the calibration utility is also able to estimate the distortion parameters of the lens. As shown in Figure 3.10, distortion can be seen towards the top left corner of the image and this is corrected for by remapping pixels from the original image using the distortion parameters.

Figure 3.10: Result of camera image undistortion using the calibration utility. Distortion can be seen towards the top-left hand corner of the image on the left.

3.2.2 Stereo Calibration

To find corresponding features extracted in one camera in an overlapping area in the second camera, they must first be stereo-calibrated to determine their relative pose. The cameras are first calibrated individually using the method described above. Pairs of images of multiple poses of the same pattern used earlier is then captured for stereo calibration.

Consider a point on the grid pattern located at point P in the real world. The following equation takes P from the world coordinate to the first camera's coordinate as mentioned previously, $P_1 = R_1P + t_1$. Similarly, $P_2 = R_2P + t_2$ is the point in the second camera's coordinate. If the transformation that takes P_1 from the first camera to the second camera is represented by a rotation R and translation t, it follows that $P_2 = RP_1 + t$. Thus, it can be show that $R = R_2 (R_1)^T$ and $t = t_2 – Rt_1$. From point extracted from the pairs of images, it becomes relatively straight forward to solve for R and T where $T = R^{-1}t$.

To prove that the calibration was successful, a rectification process is done, the method adopted here is based on work of [55]. The aim is to find a perspective transform that brings image plane of the first camera into alignment with the second camera so that they are row aligned. The rectification matrix of the first camera R_{rect1} transforms points in the camera coordinate P_1 to the new rectified coordinate P_{rect1} as show in Figure 3.11. A detailed explanation of the algorithm is provided in [52].

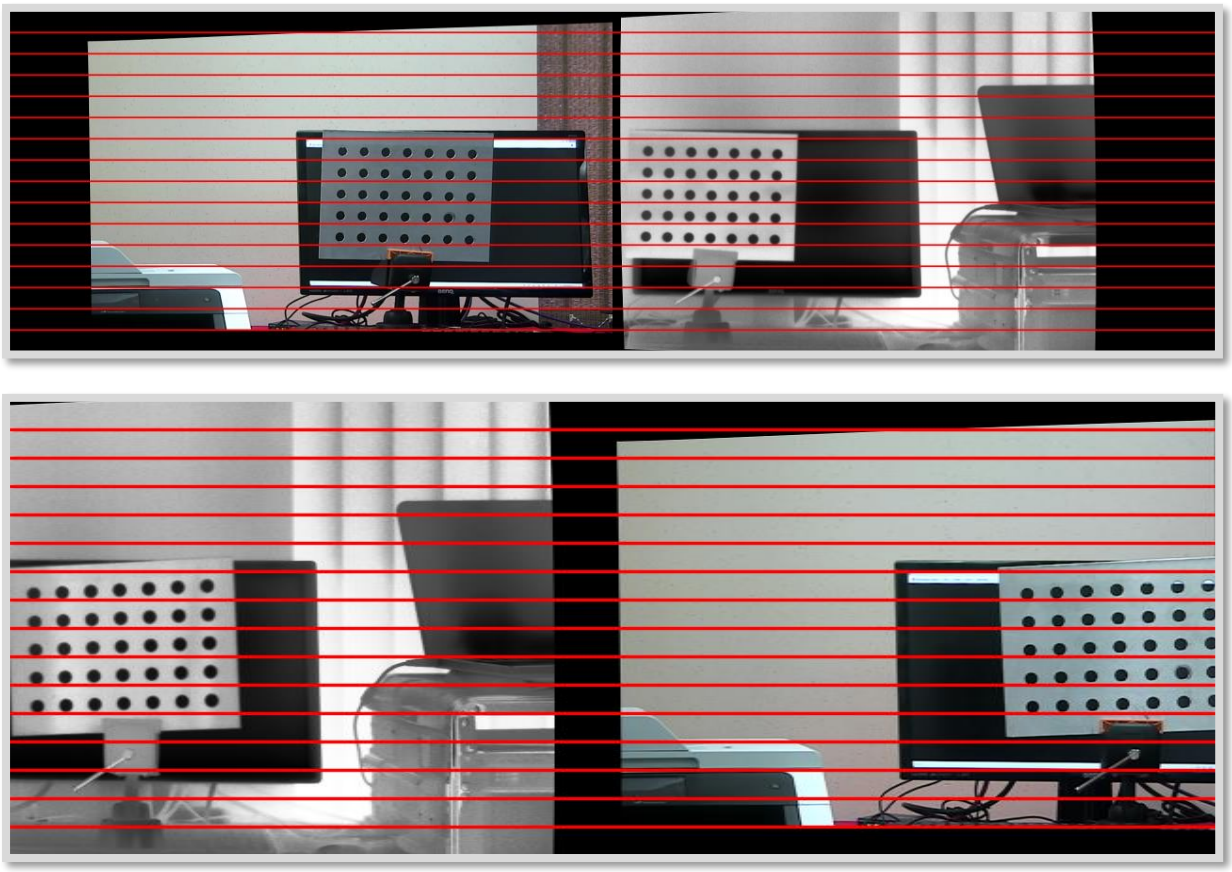


Figure 3.11: Result of stereo rectification of thermal and HD camera showing row aligned image planes. Top) HD camera aspect ratio is used; Bottom: thermal aspect ratio is used.

Note that the result of the rectification process is to make the conjugate epipolar lines parallel and row aligned. An epipolar line ($ax + by - c = 0$) in the first image plane is one where a point (x, y) in the second image plane must lie i.e. one of the numerous points that makes up the line in first image plane corresponds to the point in the second image.

The algorithm can cope with the fact that the aspect ratio of both cameras is different. The thermal and visual camera have an aspect ratio of 1.33:1 and 1.78:1 respectively. This explains why the visual camera is more warped than the thermal camera. That is, while the thermal camera rectification results in a shift of pixel to the left, the visual camera results in a shift to the right as well as being padded at the top and bottom.

3.3 Spatio-Temporal and Online calibration

Spatial alignment of multiple cameras (visual and thermal) has already been resolved by the stereo-calibration utility described in the last section, the particular case of aligning the inertial unit with a camera is addressed in section 3.3.2. In section 3.3.3, the uncorrelated issue of temporal alignment for all sensors type is addressed. But first, the coordinate notation adopted in this section is outlined in brief next.

3.3.1 Coordinate notation

In this section, the right-hand coordinate system is used, and the major coordinate frames are

- **Global frame** (G) is the earth or global frame, 3D landmark features such as the planar pattern are assumed fixed in this frame.
- **Camera frame** (C) is located at the optical centre of the camera with the z-axis pointing towards the scene, the y-axis pointing downwards and the x-axis pointing to the right.
- **The IMU body frame** (B) is attached to the body of this IMU and inertial measurements are in this frame with respect to the global frame.

Matrices (e.g. rotation or projection) are written as italic and capitals (R, P), vectors are lowercase italics with a subscript (b_k, v_G), scalars are lower case Greek alphabets (μ, α) and quaternions are 4 element vectors written in boldface italic ($\boldsymbol{q}$); readers are referred to [56] for an introduction to quaternions. The vector subscripts are also written to denote the coordinate frame in which they are measure e.g. v_C refers to a vector in the camera coordinate frame. Rotation matrices ($R_{[To][From]}$) and unit quaternions ($\boldsymbol{q}_{[To][From]}$) express a rotation that transforms a vector from one coordinate system to another:

$$v_C = \boldsymbol{q}_{CG} \circledast v_G \circledast \boldsymbol{q}^*_{CG} \qquad 3.17$$

OR

$$v_C = R_{CG}\, v_G$$

where ⊛ represents quaternion multiplication and $\boldsymbol{q}^*$ is a conjugate of a quaternion. Also, change in rotation are expressed in the global coordinate frame such that:

$$R_{CG_{t+1}} = \Delta R_G R_{CG_t} \tag{3.18}$$

or

$$\boldsymbol{q}_{CG_{t+1}} = \boldsymbol{\Delta q}_G \boldsymbol{q}_{CG_t}$$

3.3.2 Spatial alignment

Spatial alignment refers to the process of deducing the transformation operator that takes spatial measurement from one space to another. In this section, the special case of aligning IMU with a camera is described. Since the arrangement of the IMU and camera is known, the relative orientation may be estimated by analysing the coordinate system diagram of the units. For example, in the arrangement used in this thesis, we have that rotating by 90 degrees about x-axis, 0 degrees around the new y-axis and 90 degrees around the new z-axis in the Euler 321 format, the IMU coordinate can be brought in alignment with the camera coordinate. Therefore, the rotation that transforms a vector in IMU coordinate to camera coordinate is given as;

$$R_{CB} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{3.19}$$

But this assumes that the arrangement of the sensors is perfect which is not often realisable in practice. Here we employ an alternative but relatively simple offline calibration procedure to address this issue. The calibration procedure requires only the camera calibration pattern developed in 3.2.1.2. Since the IMU is rigidly attached to the camera, the relative orientation is fixed and thus only needs to be estimated once. For the algorithm described here, it is assumed that the camera has been calibrated (as described in section 3.2.1) and the IMU noise and bias has also been estimated is a separate operation. In this thesis, we employ the use of an off-the-shelf IMU that has been calibrated from factory.

The approach here follows the work of [47] whereby the unit vector corresponding to the vertical direction is observed in both sensors. This vector can be obtained directly from the accelerometer reading of the IMU and extracted from the camera by examining the vanishing point of vertical structures in the calibration pattern. By observing this vector in several poses, the relative orientation of the sensors may be deduced. Let $\boldsymbol{q}_{CB}$ be a quaternion that rotates the vertical unit vector v_B in the IMU coordinate to the camera coordinate. In an ideal scenario, the transformed vector should be parallel to that observed in the camera coordinate v_C. Since the dot of parallel unit vector in a coordinate system is one, $\boldsymbol{q}_{CB}$ can be estimated by maximizes the following equation:

$$\max_{\|\boldsymbol{q}_{CB}\|=1} \sum_{k=0}^{n} \left(\boldsymbol{q}_{CB} \circledast v_{B_k} \circledast \boldsymbol{q}_{CB}^* \right) \cdot v_{C_k} \tag{3.20}$$

Horn [57] has shown that this can be solved into:

$$\max_{\|\boldsymbol{q}_{CB}\|=1} \boldsymbol{q}_{CB}^{T}\, N\, \boldsymbol{q}_{CB} \qquad 3.21$$

where $\boldsymbol{q}_{CB}$ is obtained as the Eigen vector corresponding to the largest Eigen value of *N* [57]. Note that each observation of the vertical reference is made with the camera held static while the corresponding IMU measurement is taken simultaneously. The advantage of taking static measurement is that any temporal misalignment between the sensors will have no effect. This approach was tested, and the following rotation matrix was obtained from the estimated quaternion:

$$R_{CB} = \begin{bmatrix} -0.0450 & +0.9986 & +0.0238 \\ -0.0013 & -0.0239 & +0.9997 \\ +0.9989 & +0.0450 & +0.0023 \end{bmatrix} \qquad 3.22$$

This is equivalent to a rotation of 122.23° about an axis (-0.56, -0.58, -0.59). Using the estimated relative orientation, the IMU orientation output is used to re-project 3D points on the target pattern to the image plane. A mean reprojection error of 1.15 pixels and standard deviation of 0.8 pixels was obtained when the calibration was done using a pattern approx. 3.5m from the camera. Results presented in Table 3-1 show that this method is consistent since an average rotation of 121.998° with a standard deviation of 0.55° was obtained after repeating the calibration process multiple times. The mean square error between the verticals obtained from the camera and the estimates from the IMU was 1.0863°.

Table 3-1: Estimated relative orientation between camera and IMU with target located at multiple distances from camera

Data set	Camera distance (cm)	Rotation (degrees)	x axis	y axis	z axis	Mean sq. error (degrees)
1	3371.881218	122.23069	-0.564277114	-0.57638211	-0.59107951	0.7788
2	2427.71596	121.64005	-0.585433916	-0.57211648	-0.57441261	1.3635
3	1839.249866	122.70506	-0.565841032	-0.57891465	-0.58709602	1.1143
4	960.5716867	122.50153	-0.562109495	-0.58077050	-0.58884508	1.5215
5	427.0764192	120.91278	-0.59334971	-0.56522850	-0.57310807	0.5055
C*		122.00002	-0.574457054	-0.57466679	-0.58288692	1.0863
A*		121.99835				

C is result of data 1-5 combined and A* is the average.*

Recall that the quaternion estimated here $\boldsymbol{q}_{CB}$ transforms a vector in IMU body space to camera space. The orientation of the IMU at any given time (t) can be transformed to a vector defined by the axis-angle representation of rotation using the well-known Rodrigues formula [56]. Given the angle θ and axis of rotation e_B, the rotation vector θ_B representing the rotation of the IMU can be obtained as $\theta_B = \theta \cdot e_B$. The corresponding rotation vector in the camera body frame is obtained as $\tilde{\theta}_C = \boldsymbol{q}_{CB} \circledast \theta_B \circledast \boldsymbol{q}_{CB}^{*}$. This is equivalent to $\tilde{R}_C = R_{CB} \cdot R_B \cdot R_{CB}^{T}$. Notice the accent, which shows that the camera rotation is an estimate from

the corresponding IMU orientation. Additional measurement of the camera orientation can be made from camera images and fused with this estimate, this is the subject of following chapters.

Notice that the translation between the camera and the IMU have not been considered at all. This is because, the application considered here only requires the orientation of the camera which can be recovered without considering translation. However, it is worth showing how re-projection error is related to the distance of the scene from the camera and becomes negligible as the distance increases. The calibration was repeated multiple times with the calibration target located at multiple distances from the camera (see Table 3-1). Results shown in Figure 3.12, reveals that the re-projection error reduces with distance. Projection of points depends only on rotation as the distance tends towards infinity. In fact, in our application where the feature of interest is the horizon, it will be shown in section 4.1.2 that camera orientation can be recovered from this feature alone.

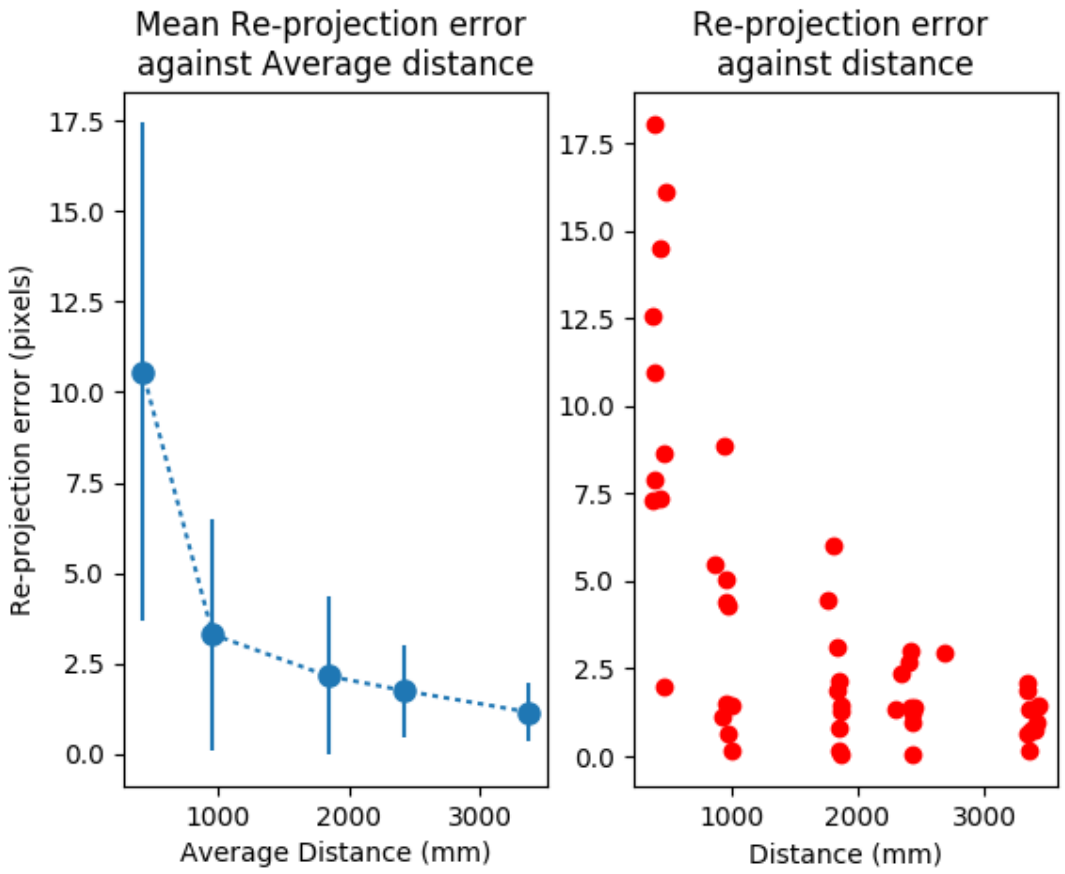


Figure 3.12: Result of reprojection error using IMU measurements and the result from spatial alignment (there are 10 samples per position)

3.3.3 Temporal calibration

To fuse measurement from multiple sensors, having a knowledge of the time difference between measurements is required to avoid potential negative consequences on the system. Not all sensors provide timestamps of measurement, some simply do not have the

capability. The strategy employed here is to timestamp the sensors data at the point of retrieval. This time is equivalent to the actual time the measurement was taken on the sensor plus some delays due time it took the sensors to process the data and transmit it to the PC, data jam caused by busy processors and jitters caused by drivers etc. [46].

The time delay between sensor pairs may be determined by online or offline calibration method, and once established this delay could be used to design a filter that fuses measurements from both sensors [58] or used to synchronize them. In the case of the latter, the sensors measurements are delayed by an amount equal to the sensor with the largest delay. This is the approach preferred and employed here. The sensors delays are estimated from an offline calibration system that follows the work of [46] and [59]. The sensors unit is placed in a repeating motion sequence and the measurement observed by sensors are recorded for a few minutes. The same pattern described in section 3.2.1.2 is placed in the view of the cameras during data capture.

The first step is to estimate the sample interval of the sensors measurements and fix any data jam and jitters as described in [46]. This is obtained by taking the average of the difference between consecutive samples; although median was used in previous work, it did not give a suitable estimate in this case. Using this estimate, excessively long difference due to data jam or jitters are detected and fixed if there are no lost packets identified; otherwise, samples caught in the jam are discarded.

Next, the magnitude of rotation of the camera is measured by assessing the pose of the calibration pattern; this requires the camera to be calibrated beforehand. A similar measurement is recovered from IMU in its body coordinate. The relative pose between the sensors is not required. The magnitude of the motion and corresponding adjusted timestamp yields a one-dimension signal that is used to align the sensors by means of cross-correlation.

For a system with *n* sensors $(n \geq 2)$, one of the sensors is selected as a reference and this process is repeated for *n*-1 sensor paired with the reference. This reduces the number of operations instead of pairing all sensors which is ${}^{n}C_{2}$ operations. The reference sensor could be any of the *n* sensors but here, it is chosen as the one with the lowest sample interval. Due to the discrete nature of the signal, the resolution of the delay estimate is limited by the sampling intervals of the sensors pair, this can be improved by upsampling all *n*-sensors data to an arbitrary sample interval *dT* using interpolation; cubic-spline quaternion interpolation is used [60]. Since the reference signal has the lowest sample interval, all sensors are upsampled to match this.

For this technique to work, the period (or frequency) of the rotation sequence employed must be at least double (or halve) that of the sensor with the lowest sample rate (or largest sample frequency) i.e. following Nyquist theorem. The cross correlation performs best when

there are several changes in direction of rotation e.g. a sinusoidal motion. Figure 3.13 shows a result obtained when this approach was tested using one camera and an IMU.

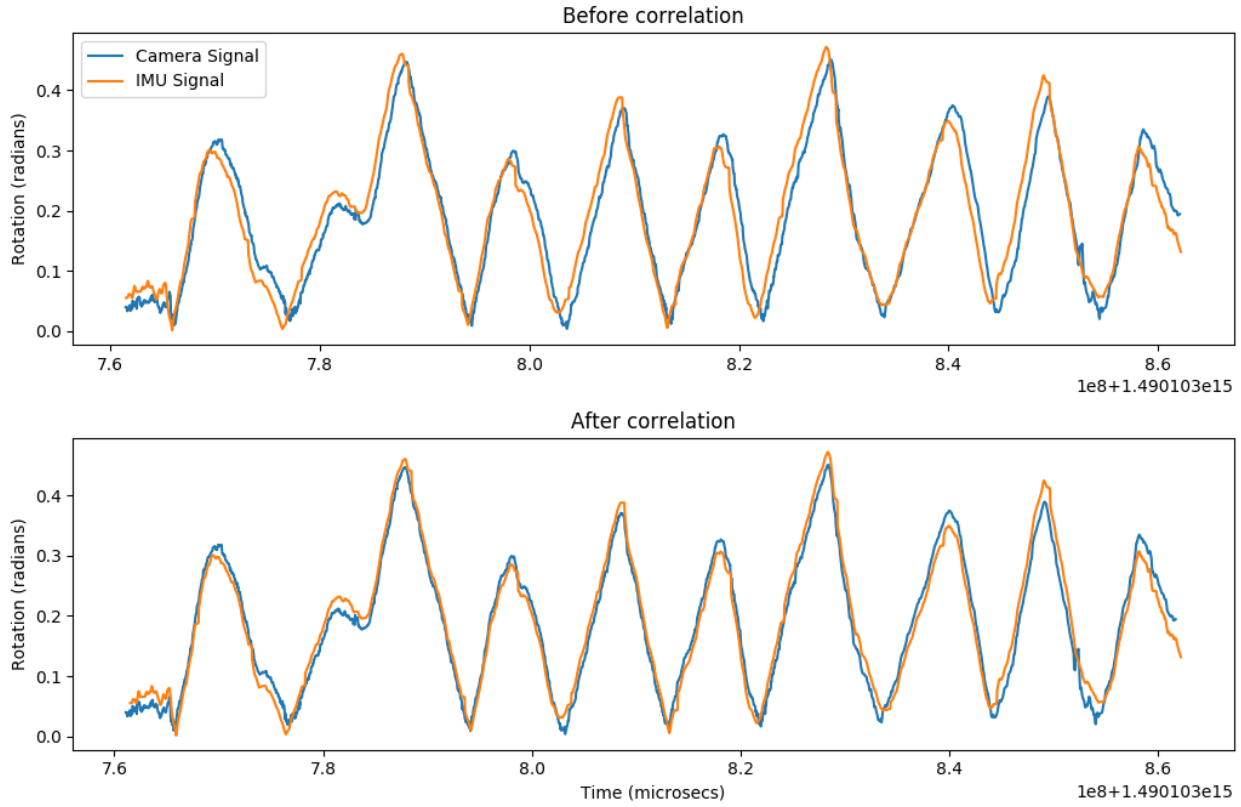


Figure 3.13: Result of temporal alignment of IMU and Camera signal

3.4 Online calibration

An important effect of the camera calibration process is that the parameters of its lens are not allowed to change otherwise the calibration is invalidated. In other words, for a zoom camera, the focus and zoom functionality of the device must be disabled. However, these are desirable features in visual monitoring application. For instance, when a target is detected, it is often useful to be able to zoom to the location e.g. to capture a high resolution image for target identification. Although the system affords digital zoom, in certain cases, optical zoom is desired.

A methodology for determining camera calibration properties regardless of zoom position is highly beneficial. A naïve implementation may require the camera to be calibrated k times for all possible combinations of camera parameters. Assuming, for example, that 10 poses of a planar object are required for calibration and the camera has 10 focus and zoom steps each, this means that 1000 poses are required; which is impractical. A number of algorithms have been proposed in literature to deal with the specific case of calibrating zoom lenses e.g. [61], [62]. However, they are still time consuming since it involves calibrating the

camera at multiple zoom positions and then fitting the result to a model to infer the calibration parameter at any position.

Sensor self-calibration (or online calibration) is a new area of research that is gaining popularity in computer vision whereby sensors are calibrated from point correspondence in a static scene viewed by a moving (rotating) camera. No calibration pattern or prior knowledge of the scene arrangement is required, thus, providing a lot of flexibility. Here, the use of horizon for sensor self-calibration is proposed. Camera calibration using a one-dimensional object fixed at one point has been investigated by [63] and extended to moving line in [64]. However, these methods require a line (1D object) of fixed length and the motion of the object must be such that it provides a constraint on all required camera parameters. Zhang [63] showed that singularity may occur if the motion does not yield the required number of constraints resulting in unreliable estimates. The horizon line does not fulfil this requirement and so a different approach is adopted.

In section 3.3.2, it was shown that the relative orientation between two sensors can be recovered by observing the vertical direction with both sensors. For a camera, the vertical direction (v_C) is related to the vanishing point (v_p) of vertical features in the view:

$$v_C = \frac{K^{-1}v_p}{\left\|K^{-1}v_p\right\|} \tag{3.23}$$

where *K* is the camera calibration matrix. It is clear from equation 3.23 that the camera must be calibrated before the vertical direction can be determined from a camera. In an image, the vanishing line (or horizon line) is made up of a set of vanishing points corresponding to parallel lines in a given plane. The relationship between the vanishing line (v_L), the vertical direction and the vanishing point is thus [65]:

$$v_p \cdot v_L = 0 \tag{3.24}$$

i.e. $$v_L{}^T K v_c = 0$$

At sea, if *N* observations of the horizon are made, it is not possible to retrieve camera parameter *K* without additional information. This is because, each observation results in *N* equations which is less than the total unknowns of 2*N* + 4. the unknowns consist of 4 intrinsic parameters that make up *K* and 2*N* orientation information related to the vertical direction which has 2 degrees of freedom; Note that translation has not been considered as explained in section 3.3.2. Here, the cooperation of visual and inertial sensor for online camera calibration is proposed since the vertical direction can be recovered from an IMU of known relative orientation.

$$v_L{}^T K \tilde{v}_c = 0 \tag{3.25}$$

where $\tilde{v}_c = \boldsymbol{R}_{CB}\, v_B$ is the camera vertical direction estimated from the corresponding IMU measurement v_B. *K* can be obtained from equation 3.25 given a minimum of four

observations using any suitable least mean square algorithm; in fact, this yields a closed-form solution. This solution can be further refined using maximum likelihood estimation by minimizing the following function:

$$\min_{r_i>0,} \sum_{k}^{N} \left\| \tilde{v}_{c_k} - v_c(r)_k \right\|_2 \qquad 3.26$$

where $r = [f_x \quad f_y \quad c_x \quad c_y]^T$

This is a non-linear minimization problem with the constraint that parameters in r are all positive numbers and the initial estimate is obtained from the closed form solution. Modern cameras tend to have a negligible skew parameter; therefore, the optimization problem can be further constrained if we let (c_y, c_x) coincide with the image centre and $f = f_y = f_x$.

Real data test

The camera and IMU data collected simultaneously for 28s and the relative orientation between the sensors was obtained previously (offline) in equation 3.22. The camera was calibrated prior to data collection using the planar method in section 3.2.1.2 and the results are shown in Table 3-2. The proposed online calibration method performed well and can cope with a significant amount of noise using the maximum likelihood approach.

Table 3-2: Online camera calibration results compared to the planar method

Solution	c_y	c_x	f_y	f_x
Planar pattern method	375.176	685.214	1306.94	1310.31
Least square method (Closed form solution)	340.853	643.098	1683.799	1884.082
maximum likelihood method	359.222	668.927	1278.015	1444.017
Constrained maximum likelihood method ($f = f_y = f_x$)	360	640	1294.793	1294.793

Simulation test

Additional experiment was conducted using simulated data; the simulated camera has the following property:

$$K = \begin{bmatrix} 2000 & 0 & 640 \\ 0 & 2000 & 360 \\ 0 & 0 & 1 \end{bmatrix} \qquad 3.27$$

The image resolution chosen is 1280 x 720 and camera motion at sea is simulated at 25fps for 30s as a sine wave of amplitude 15 and 5 degrees and frequency 0.2 and 0.5Hz in the cameras' Z and X axis respectively. Gaussian noise of zero mean and standard deviation σ was added to the orientation measurement to simulate noisy IMU measurements and errors in the relative orientation estimate between the two fictitious sensors. Note that the error in orientation measurement is equivalent to error in the localization of the left and right coordinates of the horizon line in the image and thus can be expressed in pixels. The expression for recovering horizon image coordinates from camera orientation is given in Chapter 5 (see section 5.2.2).

For each noise level, 100 independent trials were performed and the mean error in the parameter estimate was recorded. The noise level was then fixed at $\sigma = 10\ pixels$ but the camera focal parameter was modified by a zoom factor; 10 independent trials were conducted for each zoom factor. Results shown in Figure 3.14 indicates that the closed form solution seemed to have a lot of error especially in the estimation of parameter f_x, and it becomes unreliable as the noise level increases. However, the reason for this error is not investigated further since the method based on non-linear optimization gave much improved and stable results. In addition, modifying the zoom factor did not result in any notable change in relative error, thus, proving the consistency of this method.

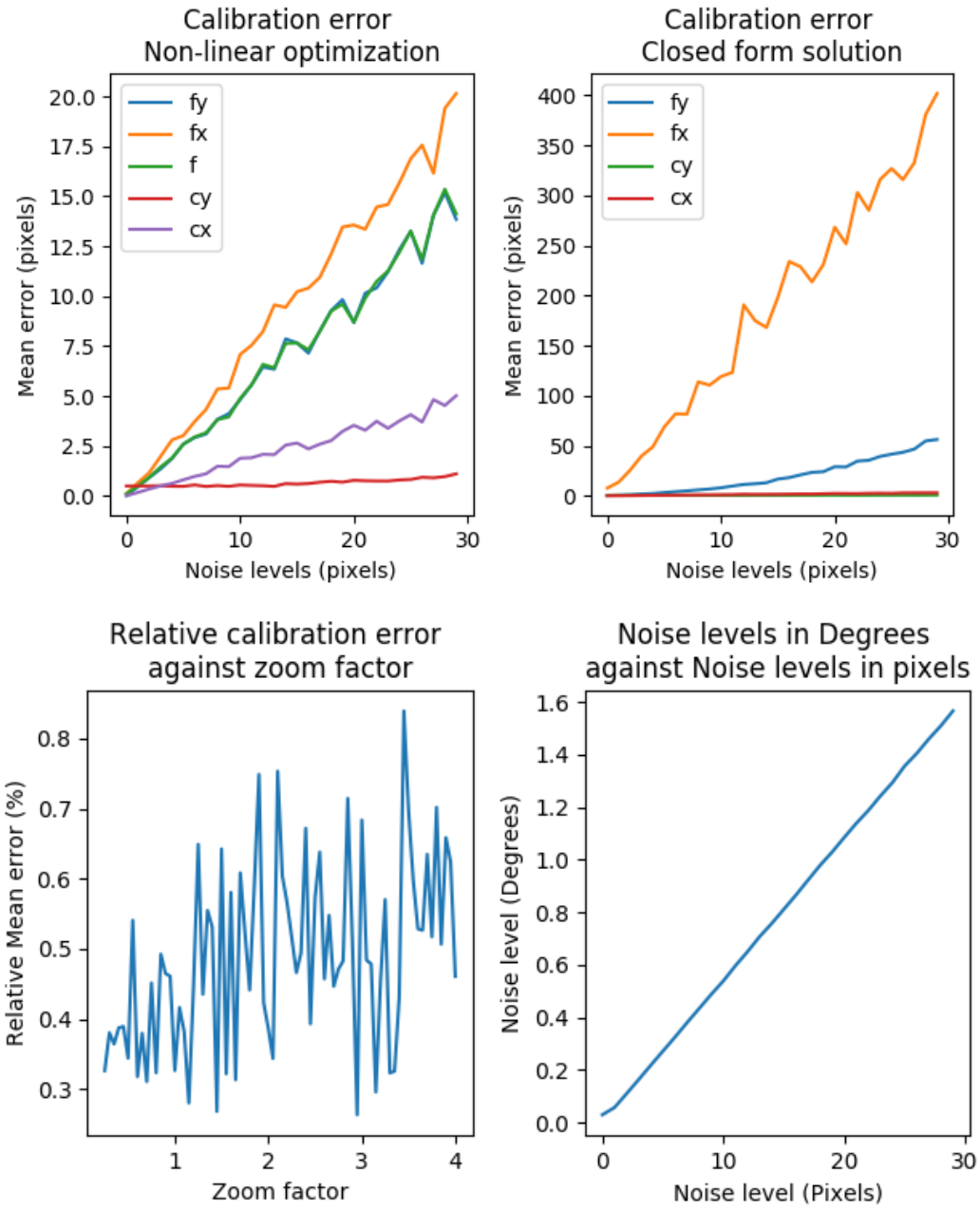


Figure 3.14: Calibration error obtained from simulated data; Top Left: Calibration error relative to noise levels from 100 independent trials using Non-linear optimization method; Top Right: Calibration error relative to noise levels from 100 independent trials using closed form solution; Bottom Left: Calibration error relative to zoom factor after 10 independent trials; Bottom Right: Relationship between noise level in pixels and noise levels in degrees.

Online spatio-temporal calibration

One of the main drawbacks of the offline spatial calibration method is that it requires the calibration pattern to be accurately positioned to reduce error in the estimation of the vertical direction. Naturally occurring features such as the horizon at sea eliminates this problem, allowing more accurate estimation of the relative orientation between two sensors.

If a camera was calibrated using the planar pattern method prior to deployment at sea, the temporal misalignment and relative orientation between the camera and the IMU can also be estimated online using the horizon line. In fact, this is simply the reverse operation of the online camera calibration process. Once this is recovered, the camera lens parameters can then be allowed to change, and camera online calibration is then performed as per normal.

Online spatial calibration is performed using the solution to equation 3.21 but in this case, the camera's vertical direction v_c is obtained from the horizon line by solving equation 3.24. Whereas, for temporal calibration, the magnitude of IMU rotation is compared with camera orientation obtained from the horizon line using the cross-correlation method introduced in section 3.3.3. The expression for recovering orientation of camera from horizon is formally derived in the next Chapter (see section 4.1.2). Finally, spatial and temporal parameters can be further refined using a maximum likelihood operation by minimizing equation 3.26, albeit, with different control parameters; i.e.:

$$\min_{||\boldsymbol{q}_{CB}||=1;\, t} \sum_{k}^{N} \left\| \boldsymbol{q}_{CB} \circledast v_{B_k} \circledast \boldsymbol{q}_{CB}^{*} - v_c(t)_k \right\|_2 \qquad 3.28$$

where t is the time delay between camera measurements and IMU measurements. The result of this is a relative orientation of 118.604° about an axis (-0.58, -0.57, -0.57); which is very similar to the expected result as obtained (offline) in equation 3.22. The estimated delay was 92.795ms and the result of the temporal alignment is as shown in Figure 3.15.

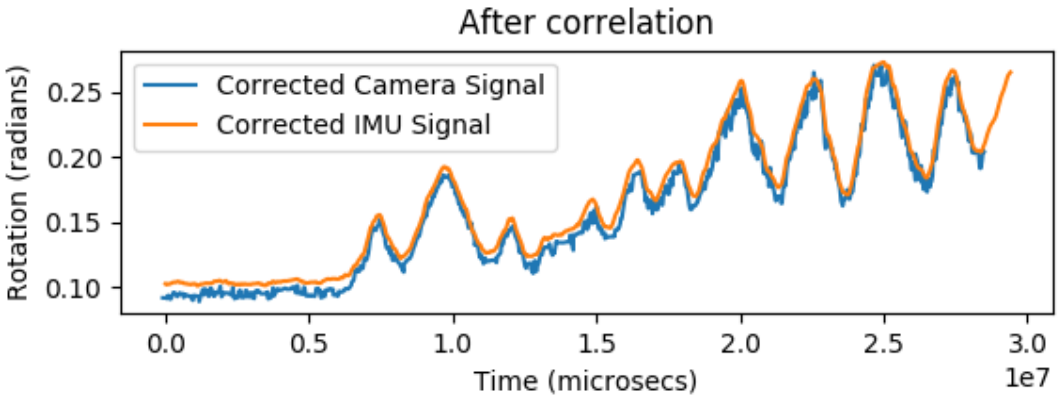


Figure 3.15: Online Temporal calibration result showing camera and IMU signal aligned in time

3.5 Summary

In this chapter, the hardware components that form the sensors unit designed for the RHVM system are described. The camera monitoring system provides the hardware framework required by the algorithms developed in the following chapters. It is used to obtain the video sequence used by the HoT system described in Chapter 5 and the ARCS algorithm introduced in Chapter 6. The various calibration utilities required for the unit has also been rationalized and duly introduced. This includes the camera calibration utility based on the pin-hole model and a planar object. The planar object is essentially a light-coloured circle grid placed in front of a dark heat source; which proves sufficient for calibrating both a thermal and visual camera and is also employed in temporal and spatial alignment of the cameras with an inertial unit.

One of the main contributions of this chapter is the introduction of a new online calibration technique. The presented solution is applicable for camera calibration as well as estimating spatial and temporal alignment between sensors. This involves the application of the horizon line detected in images at sea for calibration contrary to existing methods that require a calibration pattern. This technique is based on the cooperation of the IMU and a camera and does not require any calibration pattern. The main advantage of this is that, it allows camera lens parameters to change during operation e.g. the user may zoom in and out to investigate features of interests.

The result of the spatial calibration process shows that the sensors arrangement is not perfect and relative orientation between the sensors should not be assumed from physical arrangement alone. The spatial calibration process operates by observing the vertical direction in the sensors; for a camera, this is done by estimating the vanishing point of vertical structures in the camera frame. In the case of online calibration, the vertical direction is estimated from the horizon line alone. The relative orientation is estimated simply as the rotation that brings the vertical direction in the IMU body frame in alignment with corresponding the camera measurements.

The result of the temporal alignment shows clearly, the misalignment of measurements and this may have a negative impact if used without first synchronising them. The parameter is estimated by comparing the rotation angle measured by the two sensors using a brute force cross correlation approach. This technique proves to be reliable and does not require any knowledge of the spatial misalignment between the two sensors. Once the sensors misalignment is known (both is space and time), it becomes easy to synchronize and fuse their measurements as described in the following chapters.

In addition to temporal and spatial alignment parameters, a key output of the calibration utility are the intrinsic properties of the cameras. These are necessary to understand the 3D scene from 2D projection in the image. These parameters form the basis of the RADES algorithm described next in Chapter 4.

Chapter 4 : Real-Time Automated Distance Estimation at Sea (RADES)

Unsophisticated methods (such as using a sighting stick) often employed in distance estimation are not very accurate or precise [21]. The delay to survey operation due to inaccurate distance estimation can be very expensive and conversely, over-estimation of distance to an animal that is in the mitigation zone can have profound consequences for the animal, hence the need for more accurate estimation techniques. In this chapter, a system developed to assist visual monitoring of the mitigation zone by providing accurate real-time distance information is presented. The system employs CMS described in 3.1 coupled with a Real-time Automated Distance Estimation at Sea (RADES) software. A literature review of work that has been done on all the steps involved in the algorithm is presented and improvements made on them are stated.

RADES consists of a technique for estimating distance in an image with a camera of known global orientation. The software uses graphics to overlay an ellipse on images indicating the edge of the mitigation zone, before displaying them in real-time to a screen. So, when a marine mammal is detected, the MMO can look at the screen to accurately determine whether it is in the mitigation zone. Compared to previous photogrammetric methods discussed in section 2.2.2 that rely on manual identification of the horizon; RADES is a completely automated, real-time system and the following improvements have been made:

1. An accurate camera calibration technique based on multiple poses of a planar pattern is used to reduce error in the calculation of camera parameters.
2. Formulas used in calculation of angles are derived mathematically from first principle with no simplifying assumptions.
3. Latest computer vision methodologies including automated image thresholding techniques are used to automate the system.
4. Inertia measurement units are used to calculate and compensate for the reduction in camera height due to the vessels rolls in real-time.

In addition to the formulation of the RADES algorithm, another main contribution of this Chapter is the mathematical derivation of the resolution of the distance estimation system. This is critical to ensure that the correct sensor parameters are selected for a required level of accuracy. The RADES system relies on the horizon tracking system presented in Chapter 5 to recover vessel orientation. In section 4.1, the basic theory of the distance estimation techniques used are described and the resolution of the system is derived. In 4.2, the graphic system used to augment images in real-time by overlaying the area covered by the mitigation zone is introduced. Results obtained from sea trials of the RADES software are given in 4.3. Finally, a detailed analysis of the system is given in Section 4.4 and the Chapter is summarized in 4.5.

4.1 Distance Estimation Technique

In this section, the methods used to estimate distance on the sea surface using a camera only are described; assuming the horizon position in the image is known.

4.1.1 Using angle between a given point and horizon

A simple method of estimating distance at sea that has existed for centuries [21], [23] involves measuring the angle between the horizon and the waterline of any arbitrary point from a known height. RADES operates by reverse engineering this age-long principle and uses the similarity of the real-world angle between two arbitrary points and that formed on the image plane behind the lens of a calibrated camera by those two points. Figure 4.1 shows is pictorial representation showing the angle between the horizon and the edge of the mitigation zone.

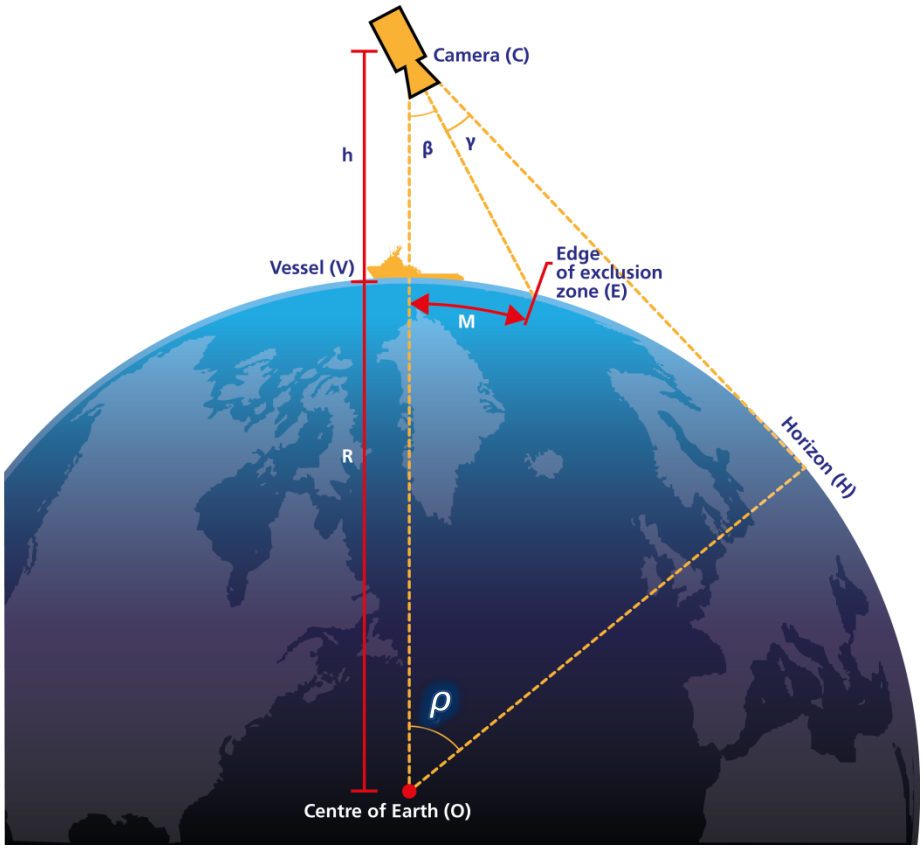


Figure 4.1: Angle subtended between horizon and waterline on the edge of the mitigation zone.

Given a point at distance *M* from the camera, real world angles can be calculated if the height of the view point is known (i.e. height of camera above sea level *(h)*). From Figure 4.1;

$$\rho = cos^{-1}\frac{R}{R+h}\ in\ radians \qquad 4.1$$

$$\alpha = \frac{M}{R}\ (radians) \tag{4.2}$$

where ρ is the angle subtended by the arc from ship to horizon as shown in Figure 4.1 and α is angle subtended by arc from ship to the edge of the mitigation zone. From sine rule:

$$\frac{\sin(\pi - (\alpha + \beta))}{R + h} = \frac{\sin \beta}{R} \tag{4.3}$$

Equation 4.3 can be simplified to:

$$\tan \beta = \frac{R \sin \alpha}{R + h - R \cos \alpha} \tag{4.4}$$

The angle γ in radians subtended from the view point between the mitigation zone and the horizon:

$$\gamma = \pi/2 - \beta - \rho \tag{4.5}$$

Compensating for Refraction of light

Equation 4.5 assumes that light travels in a straight line from the camera to the horizon. However this is not always the case, there is some terrestrial refraction [22], [23], [66]. Since the refraction of light causes the distance to the horizon to be further than it should be, this can be compensated for by replacing the radius of the earth *R* in the equations 4.1 to 4.5 with R_r:

$$R_r = \frac{R}{1 - k} \tag{4.6}$$

Where, the parameter *k*, is the ratio of the curvature of the refracted light to the earth's curvature. The value of *k* depends on temperature gradient and can be roughly estimated from it. For the rest of this thesis the value of k = 1/13 is taken as suggested in [23].

Pixel distance between horizon and exclusion zone in image

To measure angles on the image plane, the Field of View (FOV) of the camera must be known. To achieve this, the camera must first be calibrated to obtain its intrinsic parameters. RADES has an in-built calibration utility, discussed earlier in section 3.1.2 that uses multiple views (poses) of a planar object.

The ratio of the angle between two arbitrary points in the image plane to the diagonal field of view (FOV) of the camera is equal to the ratio of the pixel distance between the points and the diagonal length of the image. Since the real-world angle between the two arbitrary points is like that formed on the image plane, the result is that:

$$\frac{\gamma}{FOV} = \frac{Pixel\ Distance\ between\ horizon\ and\ exclusion\ zone\ (p)}{Pixel\ Diagonal\ lenght\ of\ image\ (D)} \quad 4.7$$

From equation 4.7, the distance between the horizon and the edge of the exclusion zone in pixels can be found. Hence, if the pixel coordinate of the horizon line is known, it is possible to locate the pixel coordinated of the edge of the exclusion zone; assuming that the roll angle of the camera is zero.

Figure 4.2 describes the operation of RADES; with the red line showing the detected horizon and the yellow line is the specified distance from the camera. The computer vision algorithm employed to automate the localisation of the horizon is described in sections 5.1.

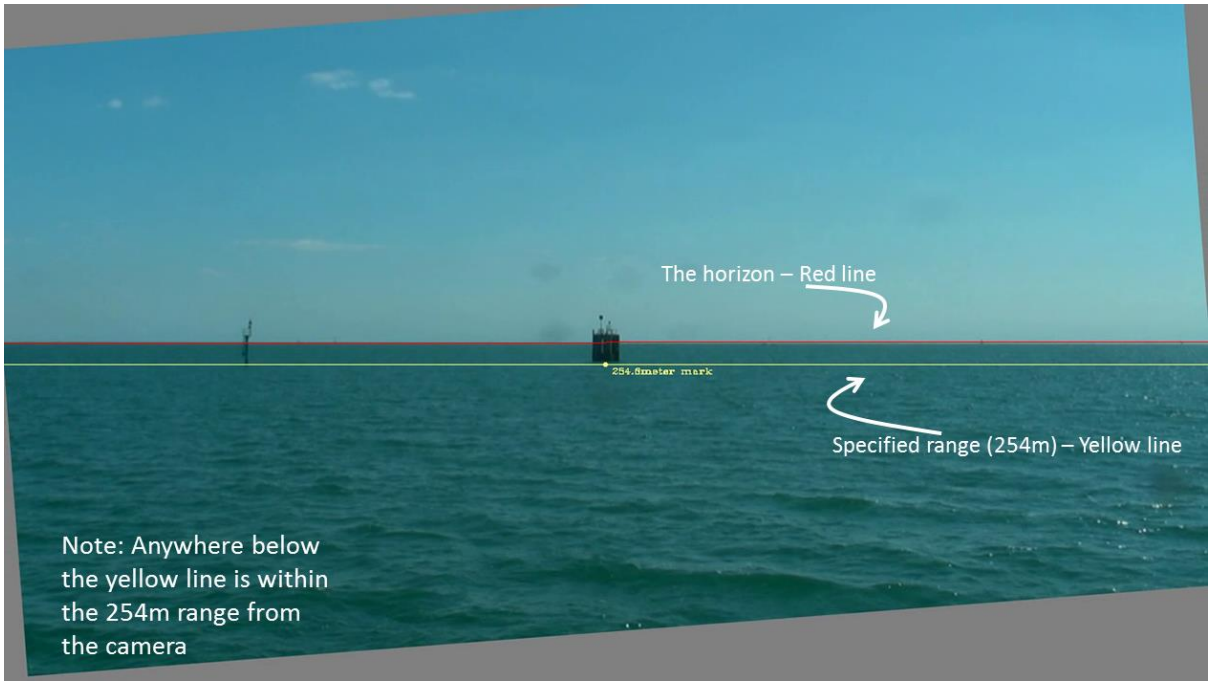


Figure 4.2: An image describing the operation of RADES. The red line is the location of the horizon and the yellow line signifies 254m from the camera.

4.1.2 Using the perspective projection equation

The method described so far enables estimation of distance if the FOV of the camera is known, for example, when it is obtained using the simple pin-hole calibration method described in section 3.2.1.1. As explained, a more robust calibration model can be used to estimate the intrinsic parameters of the camera including distortion parameters. These facilitates distance estimation using the projection equation:

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad 4.8$$

To achieve this, the global orientation of the camera must be known. Looking at the simplified scenario where translation is set to zero and the rotation of the camera about its y axis (i.e. yaw angle) is zero, equation 4.8 becomes:

$$s\begin{bmatrix}x\\y\\1\end{bmatrix}=\begin{bmatrix}f_x & 0 & c_x\\0 & f_y & c_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}c_\phi & -s_\phi c_\theta & s_\phi\, s_\theta\\s_\phi & c_\phi c_\theta & -c_\phi s_\theta\\0 & s_\theta & c_\theta\end{bmatrix}\begin{bmatrix}X\\Y\\Z\end{bmatrix} \quad 4.9$$

where θ is the rotation about the camera's x-axis and ϕ is rotation about the camera's z axis i.e. the tilt and roll angles respectively. c_θ and s_θ refer to the cosine and sine of the angles respectively.

The camera is assumed to be located at coordinate (0, 0, 0) in the real world with the y-axis pointing downwards, z-axis pointing towards the scene and positive rotation in counter-clockwise direction following the right-hand rule. Note that the yaw angle of the camera has also been set to zero because this information cannot be deduced directly from the camera. A yaw angle may be obtained from an IMU as will be discussed later. Solving the above equation and substituting *Y* equals camera height (*h)* (assuming the sea surface is completely flat) yields:

$$\frac{x-c_x}{f_x}=\frac{X\cos\theta - Y\sin\theta\cos\phi + Z\sin\theta\sin\phi}{\mathrm{Y}\sin\phi+\mathrm{Z}\cos\phi} \quad 4.10$$

and

$$\frac{y-c_y}{f_y}=\frac{X\sin\theta + Y\cos\theta\cos\phi - Z\cos\theta\sin\phi}{\mathrm{Y}\sin\phi+\mathrm{Z}\cos\phi} \quad 4.11$$

Let $[X_1,\ h,\ Z_{horizon}]^T$ and $[X_2,\ h,\ Z_{horizon}]^T$ correspond to two points on the horizon with image coordinates $[x_1,\ y_1,\ 1]^T$ and $[x_2,\ y_2,\ 1]^T$ respectively. Since $Z_{horizon} >> h$, substituting $h/Z_{horizon}$ equals zero in 4.10 and 4.11 and some simple development we have:

$$\phi = \tan^{-1}\left(\frac{f_x}{f_y}\cdot\frac{y_1-y_2}{x_1-x_2}\right) \quad 4.12$$

This shows that the roll angle is dependent on the gradient of the horizon line as expected. Further developments result in;

$$\theta = \tan^{-1}\left(\frac{(y_1-y_2)(x_1-c_x)-(x_1-x_2)(y_1-c_y)}{f_y(x_1-x_2)\cos\phi + f_x(v_1-v_2)\sin\phi}\right) \quad 4.13$$

It follows from equation 4.12 and 4.13 that, given two distinct points on the horizon line in the image, the rotation of the camera about its z-axis and x-axis can be easily recovered.

Substituting this back into 4.9, other points on the sea surface can be projected onto the image.

This method allows the incorporation of the camera lens distortion parameter estimated by the calibration algorithm, thereby further improving accuracy. It is important to stress here that the angles ρ and θ are not the same; the former refers to the angle of dip relative to astronomical horizon while the latter is relative to the geometric horizon formed by the line *CH* in Figure 4.1.

4.1.3 Resolution and reliability of image ranging technique

As shown in the previous section, camera imaging involves transforming 3-Dimensional real-world coordinates to a 2-D coordinate on the image plane; as a result, cameras are often modelled using a perspective model. Due to the perspective distortion inherent in camera systems, the pixel distance *(p)* between the horizon and a real-world point decreases as the real-world distance of that point *(M)* from the camera increases. Figure 4.3 shows a plot of *M* (meters) against *p* (pixels) for a constant FOV and image diagonal length. The legend represents the camera height in meters from 10 – 160m.

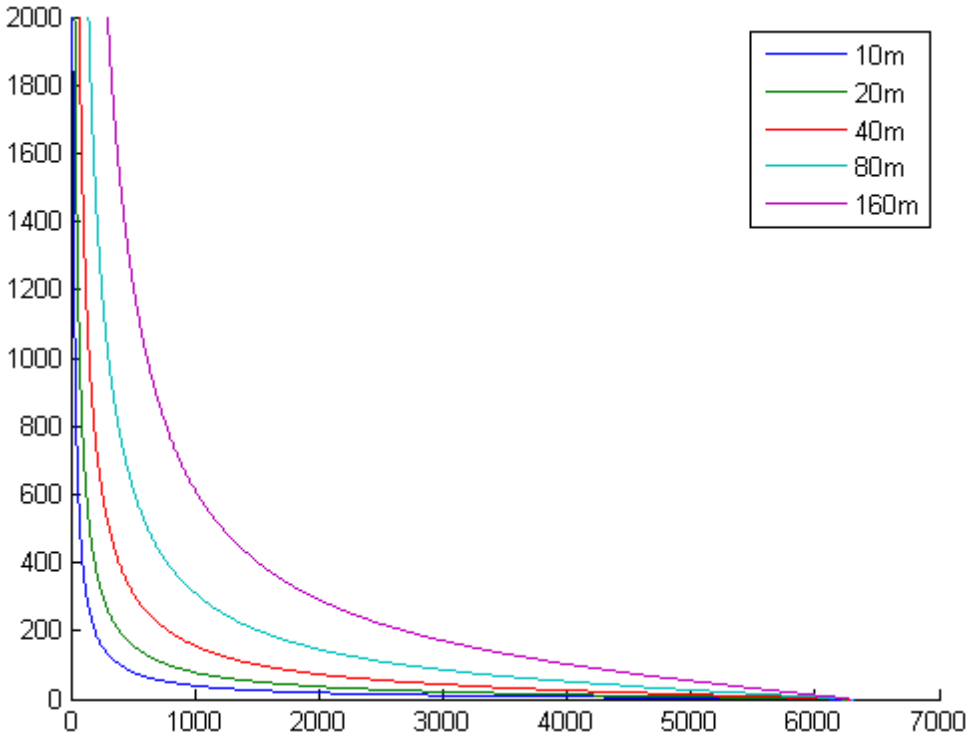


Figure 4.3: Graph of distance of real-world point (meters) against pixel distance (pixels) between horizon and that point for a FOV of 180 degrees and diagonal image length of 1280 pixels.

Results shows that the rate of change of *p* is not linear but decreases as *M* increases from zero. The consequence of this is that the resolution of distance estimates in pixels/metres also decreases as distance increases. This can also be shown in a mathematical relationship as derived next; here the generalized case described in section 4.1.1 is used for simplicity.

Combining equations 4.5 and 4.7 and differentiating the angle *(β)* subtended between the camera and the edge of the mitigation zone with respect to *p*, gives:

$$\frac{\partial \beta}{\partial p} = -\frac{FOV}{D} \qquad 4.14$$

Since *R >> M*, it is safe to assume that *sin(α) ≈ α* and *cos(α) ≈ 1*. For example, in the multi-camera sea trials (discussed later) where *M* = 1124m and *R* is radius of earth approximated as 6400km, this assumption was valid. Therefore, from equation 4.2 and 4.4:

$$tan\beta = \frac{M}{h} \qquad 4.15$$

Differentiating Equation 4.15 gives:

$$\frac{\partial M}{\partial \beta} = \frac{M^2 + h^2}{h} \qquad 4.16$$

Therefore

$$-\frac{\partial M}{\partial p} = -\frac{FOV}{D} \times \frac{M^2 + h^2}{h} \qquad 4.17$$

The parameter *dp/dM* defines the resolution of a system in pixel/meters and the negative sign denotes the fact that *p* decreases as *M* increases. Equation 4.17 can be used to calculate the maximum *FOV* required to measure a distance of *M* with a minimum resolution of *dp/dM* for a constant image diagonal length *D*, and camera height *h*.

For example, let *D* = 1280 pixels; *h* = 10m; *M* = 500m; if a minimum resolution of 0.2 pixels/meter (or 5 meters/pixel) is required, the camera must have a *FOV* less than or equal to 0.256 radians or 14.662 degrees. In other words, using these parameters, it is possible to measures distance up to 500 m with a resolution better than or equal to 5 meters/pixel. So, every 5m increase in distance from zero up to 500 m will yield at least 1 change in pixel location. By selecting the right parameters based on equation 4.17 reliable image ranging can be achieved.

4.2 Graphics Engine and Video Stabilisation

The overall aim of the RADES system is to display an augmented image to the user with graphics showing the area covered by the mitigation zone. This enables them to make a quick decision when an animal is detected in the image.

Due to the movement of the vessel (hence the camera), video images can be a little difficult to watch. The roll *(ϕ)* of the vessel can be easily determined from orientation of the horizon line (as described in section 4.1.2) and used to stabilise the video images about a fixed point in the *y* axis before display. The graphics engine achieves this by rotating and then translating the image in a single affine transform:

$$\begin{bmatrix} \alpha & \beta & \left((1-\alpha)\cdot c_x - \beta \cdot c_y\right) + t \\ -\beta & \alpha & (1-\alpha)\cdot c_y + \beta \cdot c_x \end{bmatrix} \quad 4.18$$

where, $$\alpha = \cos\phi\,;\ \beta = sin\phi; \quad 4.19$$

and $$t = y_s - \left(\frac{y_r + y_l}{2}\right) \quad 4.20$$

y_s is the row in the *y* axis upon which the horizon is fixes and typically chosen between 0 and *image-height/4,* y_r and y_l are the right and left y-coordinates of horizon in the image respectively. The image is transformed by mapping pixel coordinates using the above matrix and interpolation.

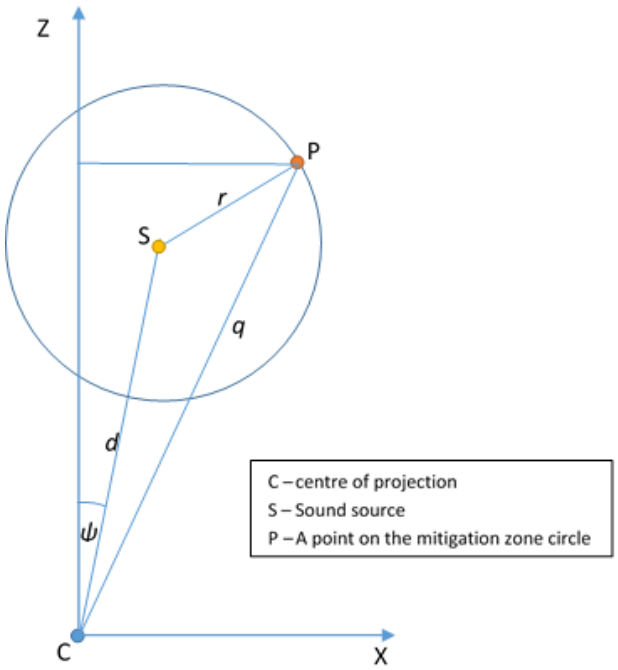


Figure 4.4: Mitigation zone of radius (*r*) about a sound source at distance (*d*) from the camera (*C*).

Finally, before display, the circle that demarks the mitigation zone must be overlaid on the image. This is achieved by sampling points on the circle. If the mitigation zone radius *(r)* is known and the distance of the camera to the sound source *(d)* is given, the $[X_P, 0, Z_P]^T$ coordinates of sample *(P)* on the circle can be obtained from the equation of a circle i.e.

$$(X-a)^2 - (Z-b)^2 = r^2 \quad 4.21$$

where, $$a = d\sin\psi\,;\ b = d\cos\psi$$

Ψ is the angle between a line perpendicular to the image plane from the centre of projection to infinity and the line from the centre of projection to the sound source as shown in Figure 4.4. The pixel location of point *P* in the image plane (u_p, v_p) can be estimated from equation 4.7 and Figure 4.4 as:

$$v_P = v_{horizon} + \frac{\gamma}{FOV} \times D \quad (4.22)$$

and,

$$u_P = c_x + \frac{\tan^{-1}(X_p/Z_p)}{FOV} \times image\_width \quad (4.23)$$

Note that *P* is also defined by its distance from the image plane *q* and the angle ϑ formed by *ZCP* (not shown in the figure). It becomes easy to obtain multiple sample point on the circle P_i *{i = 1 … n}* for every possible combination of (q_i, ϑ_i) and thus obtain corresponding location on the image plane using equation 4.22 and 4.23.

Alternatively (u_p, v_p) can be obtained from the projection equation 4.8 in a single matrix operation. This is the preferred method used by RADES, but when only the FOV is available, it falls back to the equation 4.22 and 4.23. This flexibility makes it easy for it to work with any off the shelf camera with a known FOV.

4.3 Sea trials

RADES algorithm has been tested and trailed in various setups and environmental conditions. Preliminary trials were done using a single camera system on moving and fixed platforms at two locations in the UK. A more comprehensive trial using a prototype multi-sensing system is also described below.

4.3.1 Single Camera sea trials

For proof of concept two initial trials of RADES was done using a single camera front end:

1. The first one at Portchester (50.8490893N, 1.180473W) was on a moving platform. Raw images of a fixed target at sea (a Tower Sub Barrier) were captured with the camera located on a boat. This test provided very good video data to test the stabilisation system as well as measure distance at relatively low height. Similar GPSs were used to log the positions of the target and the boat.
2. The second one at Putsborough beach (51.1243961N, 4.1987654W) had the camera mounted on land overlooking the sea capturing raw images of a Kayaker (the target). It provided useful data to test RADES at much higher heights and a bit further distances up to 920m. The same GPS was used to log the coordinates of both camera and target positions.

The height of the camera was measured using tape in Portchester while a combination of tape and theodolite at Putsborough beach. Ground truth distances between camera and target was calculated using their GPS coordinates. After data capture, the camera was calibrated to obtain camera focal length using RADES' calibration utility.

Figure 4.5: Some results from the trials at Portchester; Left: target at 254m; Right: target at 347m

Figure 4.5 shows some results from Portchester where the weather was favourable. The red line is the detected horizon while the yellow line is the distance estimated by RADES. The results show that the system is accurate since the yellow line coincides with the location of the target. The grey area in the image is as a result of image stabilisation about the horizon.

Figure 4.6: Some results from trials in Putsborough beach; Left: Original image; Right: output from RADES; Top: target at 917m; Bottom: target at 348m

Figure 4.6 shows results from Putsborough when the weather was relatively hazy. The result shows the performance of the dehazing system. The accuracy of distance estimation is further supported by the yellow line coinciding with the position of the target kayaker. Compared to the trials in Portchester, the camera was located relatively higher and then zoomed-in to improve resolution. Figure 4.7 and Figure 4.8 show some more results from Putsborough.

Figure 4.7: Result obtained with target at 578.3m

Figure 4.8: Result obtained with target at 303.4m

4.3.2 Multi-camera system sea trials

A seismic survey trial was conducted off the coast of South Africa consisting of a single vessel towing an array of air-guns and streamers (see Figure 4.9). In addition to the use of PAM and MMOs for marine mammal mitigation, for the first time, a RVHM system has also been installed on a seismic vessel.

The system installed on the vessel is a multi-camera system which consists of an array of five (5) High Definition (HD) daylight cameras (cameras 1-5), one infra-red camera (camera 6) and six computers. The daylight camera array provides a 180-degree view covering the back of the vessel, overlooking the air-gun arrays.

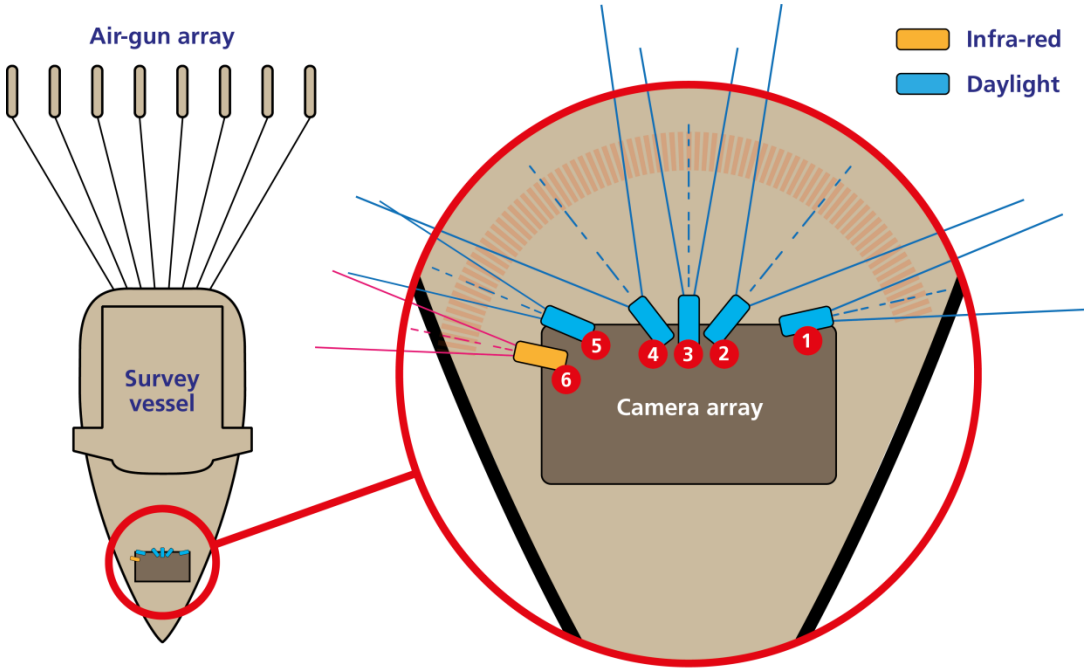


Figure 4.9: Camera arrangement on the vessel

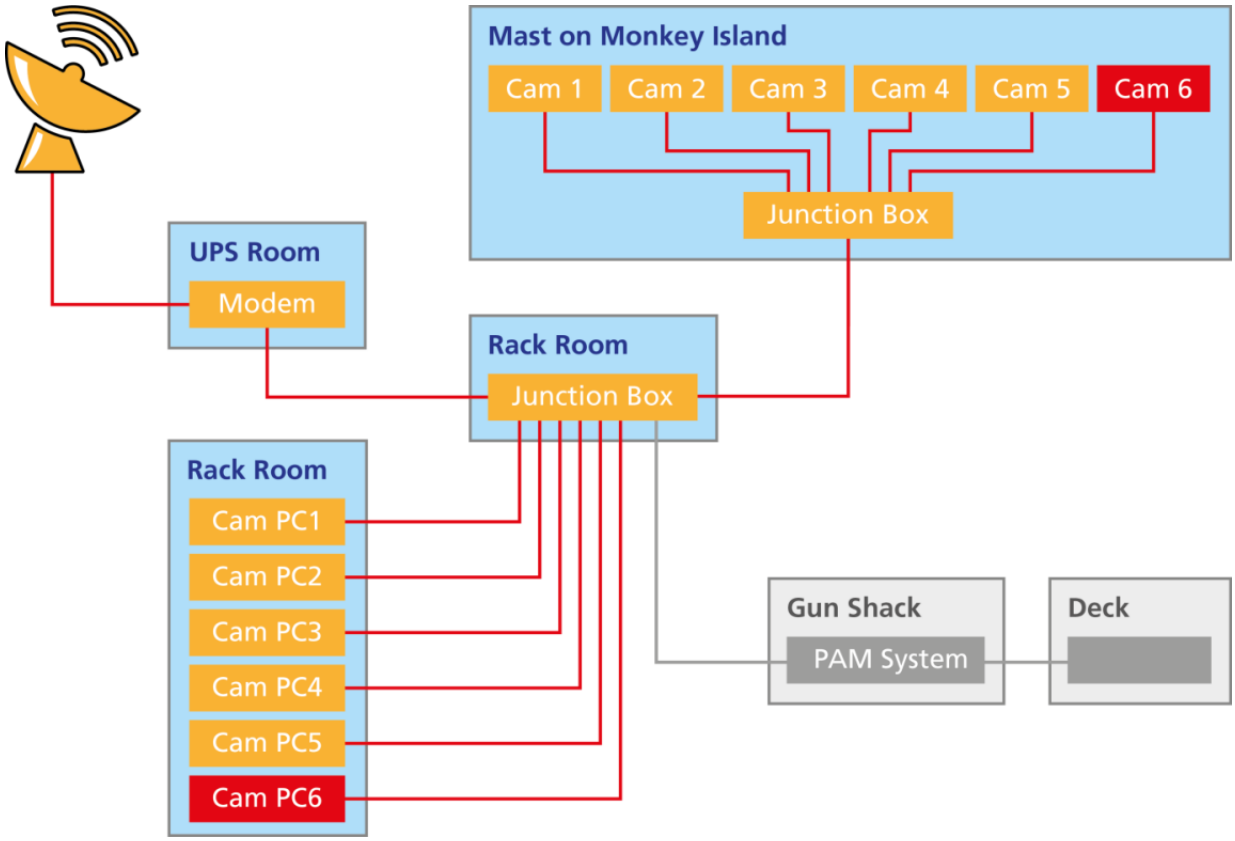


Figure 4.10: System configuration and setup on vessel.

The six cameras labelled (1-6) are mounted on a mast at a height of about 27meters above sea level. For this trial, Cameras one, three and five have a field of view of about 20 degrees while the other two are about 60 degrees. Each camera is paired with a computer

(computers one to five running RADES software). The computers are located at the instruments room located three decks below mast and they communicate with the cameras over an Ethernet network; thus, enabling visual monitoring in a relatively hazard free location. Figure 4.10 shows the system connection and setup on the vessel.

During installation, the cameras were calibrated and the results are saved on their corresponding computers. The only additional piece of information needed by RADES is the camera height and the mitigation zone distances. This information was fed to RADES by the onboard operator via its GUI before processing. Figure 4.11 shows the verification of RADES using the distance to the Air-gun array (i.e. 625m) given by the seismic crew using their highly-sophisticated GPS system as ground truth. Also, Figure 4.12, a marker buoy 400 metres away from the camera was also used to verify distance estimated by RADES.

Figure 4.11: Verification of RADES using the sound source array at 625m.

Figure 4.12: Verification of RADES using a marker buoy at 400m.

Satellite RHVM

In addition to remote monitoring from the remote location below deck on the vessel, a satellite communication link was also installed to enable onshore monitoring. The satellite system has a capacity of 512kbps and 128kps for uplink and downlink respectively and was dedicated to test remote PAM and remote visual monitoring trials.

In this configuration, all processing is done by the on-board computers and video of results are streamed back to shore using a Remote Desktop sharing software called TightVNC which is a virtual network computing (VNC) software. TightVNC uses the remote frame buffer (RFB) protocol [67] and supports varying encoding methods including Raw, CopyRect, RRE, Hextile, ZRLE and JPEG. The reader is referred to [67] if they wish to learn more about the encoding methods.

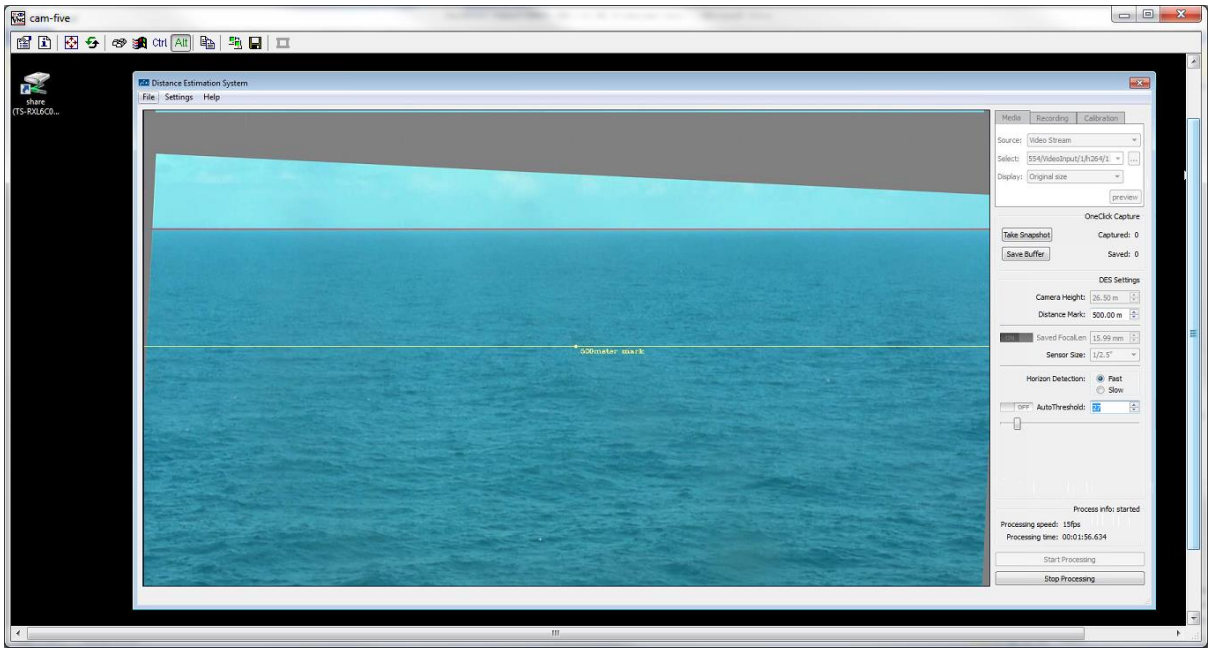


Figure 4.13: Remote High-Definition Visual Monitoring of camera 5 over satellite.

The advantage of the using desktop sharing software is that it enables the control of the RADES software and takes full advantage of its functionality from anywhere. With the software, remote monitoring was achieved using the JPEG encoding method, however, the video frame rate achievable, at reasonable subjective quality, is relatively low due to a combination of limited bandwidth and relatively high compression bit rates. However, these results prove the applicability of the system for remote visual monitoring; Figure 4.13 to Figure 4.15 shows some images from the satellite RHVM system.

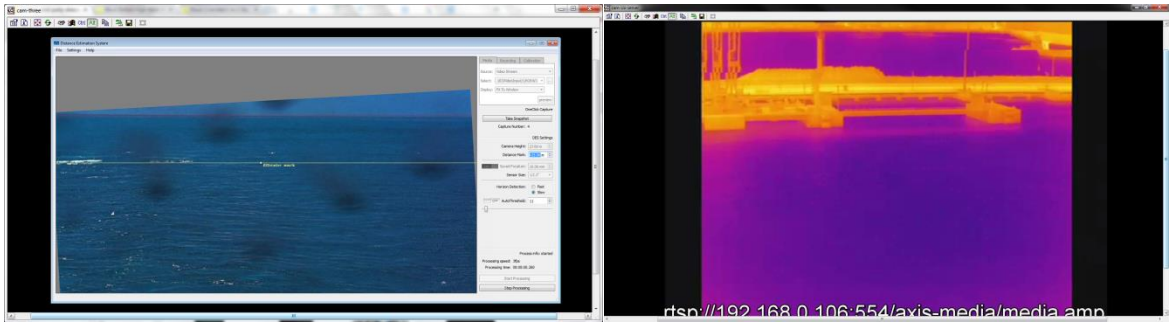


Figure 4.14: Left: Remote Visual Monitoring of camera 3; Right: Remote Visual Monitoring of infra-red camera 6.

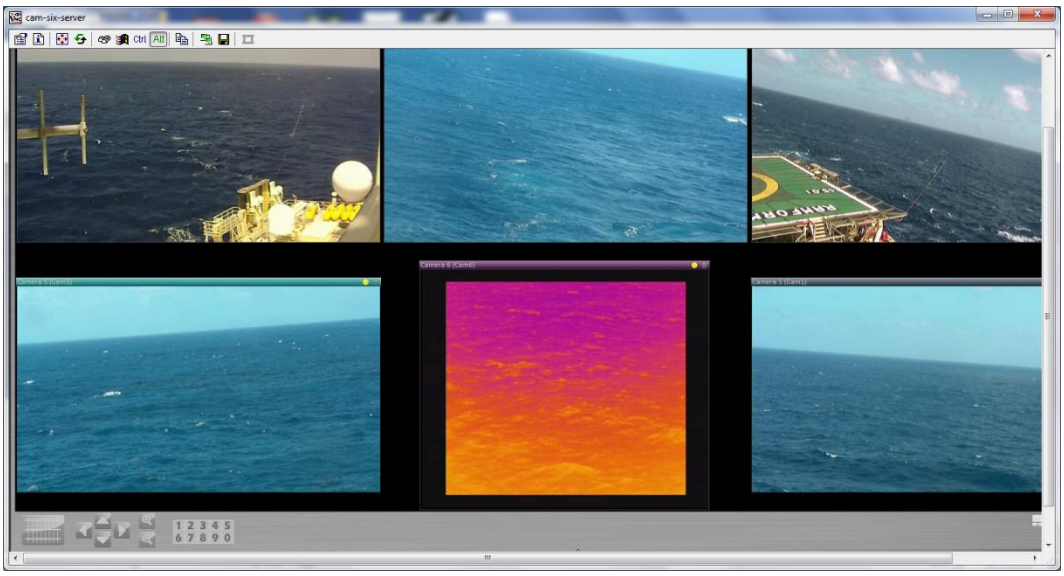


Figure 4.15: Remote Visual Monitoring of all 6 cameras over satellite using a CCTV viewing software.

4.4 Analysis of RADES system

The technique for distance estimation at sea has been described above and all the mathematical solutions required for this operation have been derived. In the last section verification of the technique and demonstrate its application in a remote sensing system was presented. In this section, a comprehensive analysis of the RADES system is presented in terms of accuracy and precision. Sources of errors relevant to precision and accuracy of estimates are considered and analysed.

4.4.1 Sources of error affecting accurate of distance estimates

During the initial tests using the single camera system, an error log was kept identifying the likely source of error that affect the accuracy of distance estimates. They include:

Table 4-1: Sources of error and description

Error	Source and description
Camera height	• Error from using measuring instruments i.e. using theodolite and tape measures. • Error due to slanting of camera based on the orientation of the boat i.e. change in orientation of boat caused by waves and swells. This results in reduction in height.

	• Error due to increase in sea level as tides changes (applies to fixed platform trials).
Camera focal length	• Camera field of view was obtained by calibrating the camera using planar patterns. Error from the camera calibration process in minimum using this technique compare to unsophisticated methods often employed.
Horizon Detection	• Slight error in the localization of the horizon due to the discrete nature of video signals and the slight shift in edges during processing. • The likelihood of this error is reduced by using high resolution HD videos and a robust edge detection technique.
Distance to the horizon	• The refraction of light causes the horizon to be a little further (or even nearer) than is calculated using geometry, trigonometry etc. This can be compensated for by increasing (or decreasing) the radius of the earth by a fraction. Accurately compensating for this is difficult since parameters needed are difficult to measure real time.

Results from the trials showed that for some tests RADES was able to provide very precise distance estimates of the target; the overall accuracy of the system was between 96-99%. Distance between the camera and targets were obtained from a GPS. The maximum error recorded was 20m. Results are promising and show that the system is good enough for research and commercial purposes.

4.4.2 Precision of distance estimate technique

As described previously, RADES uses graphics to draw a line demarking a given distance from the camera based on the position of detected horizon. The location on the image where the line is drawn is determined by the pixel distance *(p)* between the horizon and the given distance. As a result, the main factors affecting imprecision of the system can be grouped into two categories: 1) errors in the estimation of *p* and 2) errors in the localisation of horizon by software algorithms. To understand their effects, each one of these sources of error are considered separately.

Errors in the estimation of pixel distance (p)

Imprecision due to error in calculation of *p* can be determined by analysing equation 4.7**.** It can be immediately seen that the parameter *D* is a constant and thus, not considered further. Since γ is obtained from calculations derived from first principle with no simplifying assumptions, analysing equations 4.1 to 4.5, it is obvious it is only affected by errors in the measurement of camera height and FOV.

If the camera is mounted on a stationary platform with a fixed height and thus error free; errors in distance estimate will be dominated by error in the FOV since it is obtained from human calibration. Based on this assumption, we have:

$$p + e_p = \frac{\gamma D}{FOV + e_{FOV}} \quad 4.24$$

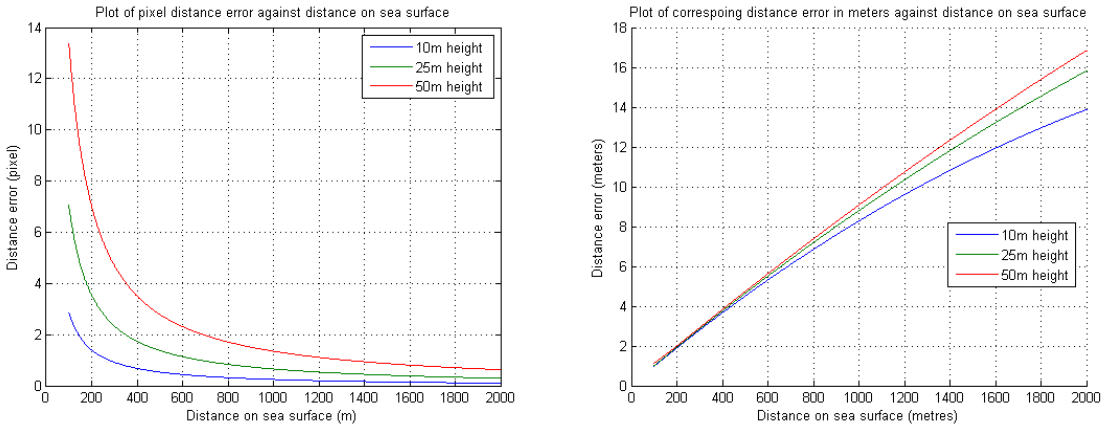
where e_p and e_{FOV} are the errors in p and FOV respectively. Rearranging equation 4.24 gives:

$$e_p = -\frac{e_{FOV}}{FOV + e_{FOV}} p \quad 4.25$$

This shows that e_P decreases proportionally with FOV. To demonstrate, multiple calibration results of the same camera was performed, a mean FOV of 0.427704634 radians was obtained and a standard deviation of 0.004196475 radians. Using these values as FOV and e_{FOV} respectively in 4.25 a plot of e_P against distance on sea surface (M) in meters is shown in Figure 4.16a.

Now, since the assumption of fixed height made earlier is not always the case, its effect on e_P can be observed by varying the camera height used in 4.25. Figure 4.16a shows the effect of height on e_P with varying camera heights of 10m, 25m and 50m. This shows that e_P increases with camera height.

Since e_P is an estimate of pixel difference between real and obtained, equation 4.17 can be used to estimate the corresponding error in meters (e_M) caused by e_P. A plot of e_M against M is also shown in Figure 4.16b. The result shows that, the pixel error e_P is greater at shorter distances and reduces as a tangent function with distance. However, the consequent error in meters at longer distances is much greater. For example, at 100m for the 25m high camera, e_P was estimated at 7 pixels but this translates to an e_M of 1meter while at 700m, e_P is estimated at 1 pixel but translates to 7m of error.



(a) (b)

Figure 4.16: Left: Plot of pixel distance error against distance on sea surface; Right: the corresponding error in meters against distance on sea surface

Errors in the localisation of horizon by software algorithms

The horizon detection algorithm is introduced in chapter 5, however imprecision in the localisation of the horizon have important consequences on the distance estimate. By defining errors e_{yL} and e_{yR} in the left and right coordinates of the horizon $(y_l + e_{yl}, y_r + e_{yr})$ respectively, this will yield a combination of angular and displacement errors about the

centre of the horizon line $(x_{horizon}, y_{horizon})$. The displacement error is only in the y-axis since the centre of the horizon line in the x-axis is always the same. Thus:

$$y_{horizon} + e_y = \frac{y_L + e_{yL} + y_R + e_{yR}}{2} \quad 4.26$$

$$e_y = \frac{e_{yL} + e_{yR}}{2} \quad 4.27$$

Also,

$$\tan(\theta + e_\theta) = \frac{y_R - y_L + e_{yL} - e_{yR}}{image\ width} \quad 4.28$$

In this thesis, $y_R - y_L$ cannot be greater than the image width, i.e. image height is always less than image width; therefore, $-1 < tan\vartheta < +1$. Assuming e_ϑ is very small, from 4.28, using small angle approximation:

$$\tan e_\theta = \frac{e_{yR} - e_{yL}}{image\ width} \quad 4.29$$

Since parallel lines are kept parallel after affine transformation, it follows that the only additional errors that may be added by the image stabilisation process are only due to quantisation because of the discrete nature of digital images. This results in additional displacement error in the y-axis (e_{qy}) and angular error about centre ($e_{q\vartheta}$) with variances $s_{qy} = \Delta_y^2/12$ and $s_{q\vartheta} = \Delta_\vartheta^2/12$ respectively, where Δ_y and Δ_ϑ are resolution in y-axis and θ-axis respectively. Using typical values of 1 and arctan (1/image width) for Δ_y and Δ_ϑ respectively, e_{qy} and $e_{q\vartheta}$ can be obtained as standard deviation of 0.289 and 2.256e-4 respectively for fixed image width of 1280 pixel.

The above analysis shows that error due to localisation is independent of camera height but solely on the resolution of the image and the image processing algorithms employed. A typical estimate of error e_{yL} and e_{yR} was obtained by calculating the standard deviation of the estimates of the horizon coordinates viewed from a stationary camera. e_{yL} and e_{yR} were estimated as 1.438 and 1.433 pixels respectively.

Analysis in the last section confirms that distance estimation using the described approach is most sensitive to the height and field of view parameters providing that the horizon detection software can consistently and accurately detect the horizon. As described earlier, RADES has a simple but accurate calibration utility for estimating the camera's intrinsic parameters and hence FOV. The error using this method is minimal. Further to this, sensors can be adopted to reduce the error in height; this is part of the plan for future work discussed in chapter 7. As the vessel pitches and rolls, the height of the camera is reduced by a length proportional to the cosine of the angle between the original vertical position and the new slanted position. The value ϑ is selected as the pitch angle e.g. from the IMU sensor since it results in change in height in a plane perpendicular to the horizon. Thus, the new camera height may be obtained as:

$$h_{new} = h_{original} \cdot \cos\theta \quad 4.30$$

4.5 Summary

In this chapter, the technique for distance estimation on the sea surface was introduced and all mathematical equations used have been derived from first principles. Two algorithms have been given to cover all possible scenarios 1) the perspective method when the camera has been fully calibrated and 2) a second method that requires only the camera's FOV given by the manufacturer. The method of recovering vessel global motion from horizon position was developed and a formula for establishing the resolution of the algorithm given the system parameters has also been derived.

The recovered orientation provides several benefits including digital image stabilisation in two axes and in the online camera calibration procedure described in previous Chapter. In addition, it facilitates the fusion of camera measurement with other sensors that measure orientation (such as an IMU) using any suitable orientation filters. The orientation filter adopted in this thesis in presented in Chapter 5.

The output of the RADES system is the overlay of graphics demarking the mitigation zone in an augmented reality style application. Analysis shows that the method is effective, and the accuracy is sufficient for commercial and research use. This is verified by results from single and multi-camera trials also presented. The main sources of error are due to height of camera and camera parameters such as the field of view. RADES relies on the calibration utility described in Chapter 3 to help deal with some of these.

RADES can be highly beneficial to MMOs for precise distance information because it is real-time and visually available on a screen. In addition to this, it allows the capture of videos and pictures as evidence of marine mammal sightings. This has the potential to improve documentations of sightings and providing irrefutable proof of a detection inside or outside the mitigation zone. In addition to the above merits, RADES has application is other unrelated field and could be used to estimate distance to small nearby vessels which can be difficult to detect in a Radar system [68].

Although remote monitoring was somewhat hampered by limited bandwidth and the inferior performance of video encoding technique used in the remote desktop software, results demonstrates the potential of the system for use in hazard free monitoring from remote locations onshore. Furthermore, the aforementioned problem can be improved upon using newer technologies such as the hybrid VNC system introduces in [51].

Finally, RADES relies on the localisation of the horizon to determine vessel orientation for image stabilisation and distance estimation. The computer vision technique developed for real-time detection of the horizon is described next in chapter 5.

Chapter 5 : Horizon Tracking (HoT) System

RADES (described in Chapter 4) relies on a combination of computer vision methodologies to find the horizon pixel coordinates and continually track it as the vessel moves. The Horizon tracking system, based on the well-known Kalman Filter enables prediction of the next position of the horizon thereby simplifying the detection process. This critical step improves processing speed and facilitates real-time processing of the video stream. In addition, RADES uses the position of the horizon to stabilize the video sequence on a fixed position hence eliminating the need for expensive video stabilisation gimbals and in some cases, it augments the performance of such gimbals.

Figure 5.1 shows a block diagram of the HoT system. Input RGB image is grabbed at regular intervals from the camera and processed. The first stage is a pre-processing step that transforms the RGB image to a dark channel image (discussed in section 5.1.1), upon which edge detection is performed to produce an edge map. The edge map is then searched for features that best describe a horizon. Using the coordinates of the horizon, the graphics engine draws the mitigation zone circle (ellipse) on the input picture before display to screen; pixel position of points on the mitigation zone are obtained from techniques described in section 4.1. In addition, the cooperation of multiple complementing sensors is exploited to further improve performance of the HoT system.

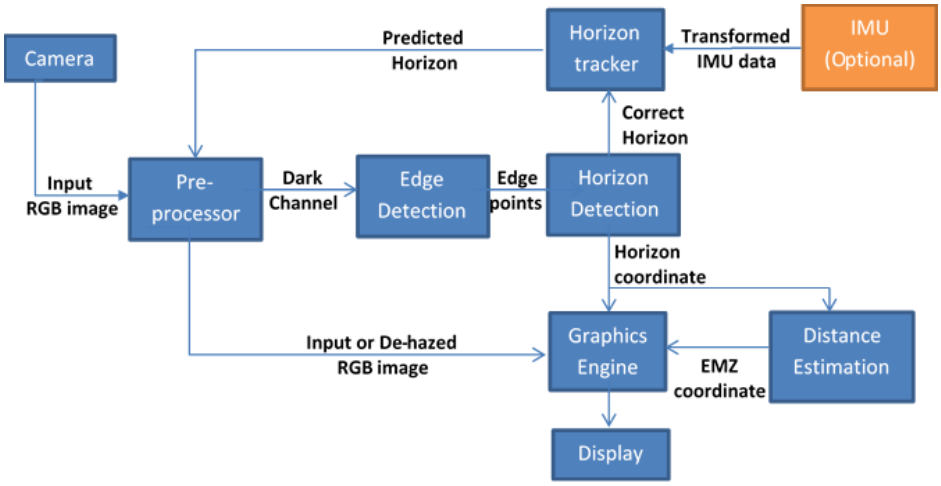


Figure 5.1: Block diagram of the distance estimation algorithm.

The uniqueness of the system comes from the innovative application of several image processing algorithms combined to form a highly intricate but effective system. Compared to previous work, the system is capable of coping with widely varying environmental conditions at sea. Horizon tracking using camera images only is described in section 5.1. The use of multiple sensors to deal with temporary occlusion of the horizon in the camera is described in section 5.2. An evaluation of the HoT system and summary are given in section 5.3 and 5.4 respectively.

5.1 Visual Horizon Detection and Tracking System

The horizon detection algorithm is the most important part of the RADES system because of its significance in the system as analysed in previous chapter. Hence the need for a detection system suitable for the marine environment where visibility is usually poor due to spray, mist and fog.

A lot of work in the computer vision field has been directed towards horizon detection especially for attitude determination in Unmanned Air Vehicles (UAV). The most common methods are based on classifying pixels into sky and ground pixels. Boroujeni et al. [69] exploited the property of dominant light field between sky and ground region and found the dominant light field using intensity-based k-means clustering. Fefilatyev et al [70] used machine learning techniques for classification in their detection algorithm and reported an accuracy of 90-99%. Ahmad et al [71] developed an edge-less method based on machine learning which has the advantage of being able to extract a continuous horizon line (i.e. without gaps). A more recent technique involving the use of deep neural network was presented in [72]. While these methods were shown to be robust, their application is significantly different from this one and the effect of environmental conditions at sea have not been considered.

For the particular problem of horizon detection in the sea environment where it is very often hazy, Libe et al [73] have reviewed the application of four well-known algorithms for detection at sea. The algorithms were implemented and applied on still sea images; they found the most accurate in terms of angular error to be based on Canny edge detector and Hough transform. Although, results presented in the paper were on images with little or no haze. In [74] and a similar horizon detection technique based on Canny edge detector and Hough transform was used for a ship based application, however their technique was used offline and no pre-processing step was adopted. In [75] multiple channels of a coloured image are processed independently with edge-preserving morphological filter and Sobel operator; the resulting edge map are then fused before Hough transform is applied. While Hough transform is robust to noise and partial occlusion, it is a relatively slow method since it uses a vote accumulation approach. In addition, processing of multiple channels further adds to the computational requirement.

Yuan et al. [76] have developed a horizon detection algorithm for foggy areal images based on an energy function dark channel space. Their system was designed for use by an UAV without any focus on environment. Although, there was substantial pitch angle error, the system was reported to be quite robust in foggy weather. The dark channel approach is a particularly interesting one because of the application of dark channel in haze removal. Qi et al. [68] have also developed a horizon detection method for the sea environment based on wavelet transform singularity analysis in the dark channel space. The technique is designed for real-time application and the approach considers the weather effects on the detection algorithm.

In this thesis, a multiresolution analysis approach based on wavelet transform was adopted for horizon detection because of its flexibility and ability to cope with noise. A detection system based on undecimated (otherwise called A Trous) wavelet transform in the dark channel space has been implemented and tested. The pre-processing and the edge detection step, described next, are precursor to the main operation of horizon detection. However, the success of the HoT system is very much dependent on the result of these steps. It is worth mentioning here that single images are used to show the principal approach, however, results using video sequence are presented later in the chapter.

5.1.1 Pre-processing

Several coloured edge detection methods exist in literature but they are, arguably, generally slower since at least 3 channels of information must be processed e.g. in [75]. Most often, the 3-channel image is transformed to a single grey image. The method adopted in here involves transforming a 3 channel RGB image into a single dark channel.

Dark Channel Prior

The dark channel prior is based on the observation that, except for the sky region of an outdoor haze free image, the intensity of the dark channel of that image is low and tends towards zero. The prior has been verified statistically [77] and used for single image de-hazing and depth mapping. This prior is valid for sea images as well because of low intensity pixels due to shadow cast by waves, swells etc. below the horizon.

Figure 5.2: Left to Right: The intensity of an image and its dark channel using 15 x 15 kernel

The dark channel involves finding the minimum intensity pixel within a local patch. The dark channel J^{dark} of an arbitrary image J is defined in equation 5.1, where c is the colour channels of the image in the RGB space taking values r, g or b and Ω is a local patch about a pixel x. [77].

$$J^{dark}(x) = \min_{y \in \Omega(x)} \left(\min_{c \in \{r,g,b\}} J^c(y) \right) \qquad 5.1$$

Computing the dark channel of a sea image is particularly useful because it acts like a filter for smoothing noise due to waves while still enhancing the sky-sea boundary. In addition, since the output is a single channel image, computational requirement of subsequent steps

is significantly reduced. In situations where the image is too foggy for visual monitoring, images can first be de-hazed before display on the screen.

In the algorithm used here a 'disc' kernel is used instead of the rectangular one used in literature. This is because, as will be shown later, artefacts introduced by the dark channel are easier to deal with when de-hazing; since most naturally occurring objects generally tend to have curved rather than boxed edges. Figure 5.2 shows the intensity of an image and the dark channel using a 15 by 15 kernel. The kernel size is important because the bigger the size, the more the chances of the kernel containing a dark pixel [77]. It can be seen that the horizon-sky boundary is clearly enhanced in the dark channel.

Haze Removal using Dark Channel.

The dark channel prior is useful for haze removal [77] and depth map estimation [68], [77], [78]; when the prior is fulfilled, the transmission of light can be easily estimated from the widely used image formation model:

$$\boldsymbol{I}(x) = \boldsymbol{J}(x)t(x) + \boldsymbol{A}(1 - t(x)) \tag{5.2}$$

Where ***I*** is the hazy observation of the original image scene ***J*** with a global atmospheric lighting ***A*** and medium transmission *t*. Rearranging equation 5.2 and taking the dark channel of both sides gives:

$$\min_{y\in\Omega(x)}\left(\min_{c\in\{r,g,b\}}\frac{I^c(y)}{A^c}\right) = t(x)\min_{y\in\Omega(x)}\left(\min_{c\in\{r,g,b\}}\frac{J^c(y)}{A^c}\right) + 1 - t(x) \tag{5.3}$$

since,

$$J^{dark}(x) = \min_{y\in\Omega(x)}\left(\min_{c\in\{r,g,b\}}J^c(y)\right) = 0 \tag{5.4}$$

$$t(x) = 1 - \min_{y\in\Omega(x)}\left(\min_{c\in\{r,g,b\}}\frac{I^c(y)}{A^c}\right) \tag{5.5}$$

Atmospheric light ***A***, which corresponds to the colour of the most haze opaque pixel, is estimated as the brightest pixel in the dark channel with the highest intensity in original input image ***I*** [77]. Using the estimated transmission *(t)*, the original version of the observed/input image can be recovered from equation 5.6. The transmission is restricted to a lower band t_0 due to potential noise that may be introduced when *t(x)* is close to zero [77]. Hence;

$$J(x) = \frac{I(x) - A}{\max(t(x), t_o)} + A \tag{5.6}$$

However, halos and block artefacts are introduced since the transmission is not always constant in a given patch [77]. As shown in Figure 5.3, artefacts can be seen across the horizon and around the kayaker.

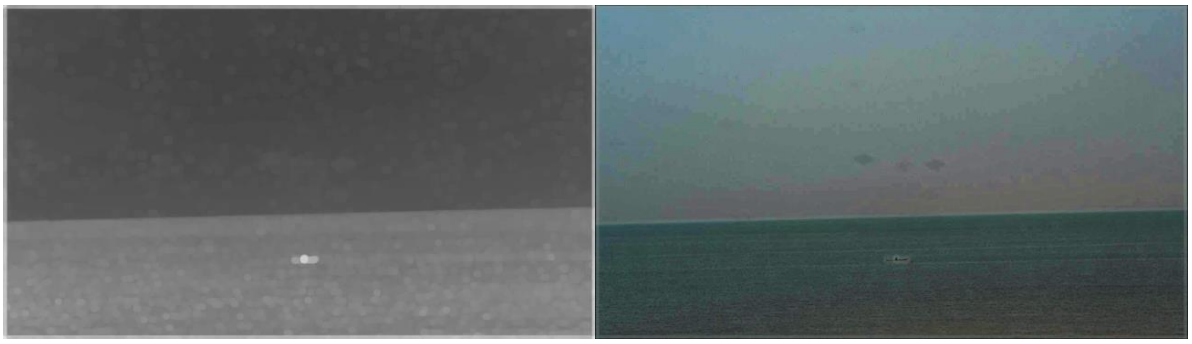


Figure 5.3: Left to Right: Transmission map and result of dehazing.

Consequently, soft matting is used to refine the transmission map since the haze equation is similar to the soft matting equation [77]. This has the disadvantage of being very slow for real time operation. A number of alternative methods for refining the transmission map have been developed. For example a guided image filter that can be used for refining the transmission map has been developed [79] and a bilateral filter has also been used [80]. The bilateral filter is still a relatively slow process and approximations used to improve speed can be inaccurate [79]. In the next section, two very simple alternative methods of refining the transmission map similar to that described in [78] and [81] are implemented and compared.

Improving Haze Removal

In the first method, the transmission map is refined using basic morphological operation of erosion and dilation. A disc kernel is used in the computation of the dark channel since most naturally occurring objects have curved edges. Then the transmission map is eroded with a cross kernel of the same size as the disc kernel. In the second method, transmission is estimated using a variation of the dark channel prior based on median filtering introduced in [81]. In this method, transmission is estimated as:

$$t(x) = 1 - \underset{y\in\Omega(x)}{\text{median}}\left(\min_{c\in\{r,g,b\}} \frac{I^c(y)}{A^c}\right) \quad 5.7$$

The advantage of the median dark channel prior is that it is faster and refining of transmission map is hardly required [81]. In both cases, just like in [77], a weighting constant is used to retain some haze in the image so that it continues to appear natural after dehazing, hence a modified transmission:

$$\tilde{t}(x) = 1 - \omega(1 - t(x)) \quad 5.8$$

(c) (d)

Figure 5.4: (a) original image; result of haze removal when (b) transmission map is modified using morphological operation; (c) using median dark channel approach (d) using the approach in this thesis.

Figure 5.4 (b) and (c) shows the result obtained from both approaches using ω of 0.85 and t_0 of 0.3. Results show that the erosion approach does not completely remove all artefacts in the transmission map hence introducing undesirable noise in the image during further processing. On the other hand, the transmission map estimated by the median dark channel is too dark and hence the recovered image scene is dark with halos and artefacts introduced at the bottom of the image. The removal of haze using the median dark channel variation results in a very dim image. This is because the transmission estimated small [80] hence the need to further refine the transmission map by adding a constant p as in Equation 5.9.

$$\tilde{t}(x) = 1 - \omega\big(1 - t(x)\big) + p \qquad 5.9$$

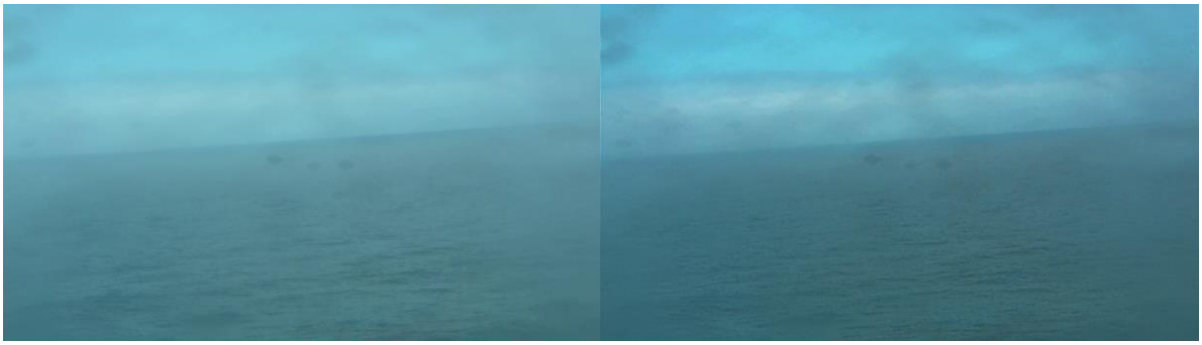


Figure 5.5: Left to Right: original image and result of haze removal using the median dark channel and brightness correction approach adopted in this thesis.

Haoran et al specified p in a range between 0.08 and 0.25 [80]. Using p = 0.25, the recovered image does not appear as thorough but does not contain as many artefacts and produces a much better visual result as shown in Figure 5.4(d). All approaches operated at similar speed and are much faster than the use of soft mapping. The brightness correction approach for

dehazing using median dark channel is the method adopted in this thesis. Figure 5.5 shows another result using the adopted method.

Upon dehazing, the new dark channel of the image can then be recomputed for further processing. Figure 5.6 shows the dark channel before and after dehazing and the scaled difference between them. The difference shows that the sky/sea boundary has been enhanced by the dehazing process.

Figure 5.6: Left to right: Dark channel before haze removal; Dark channel after haze removal; and the scaled difference (original image is shown in Figure 5.3).

5.1.2 Edge detection

Edge detection is one of the most important aspects of image processing and it is usually an intricate step towards extraction of features. Edges refer to boundaries where there is a change in intensity level or colour and they can be found by examining pixel neighbourhoods for these changes. In fact, traditional edge detectors such as Sobel and Prewitt used in grey scale images are essentially gradient operators that find edges by examining a pixel's neighbourhood. Coloured edge detectors are also becoming more popular with increase in processing power of modern day processors and a comprehensive comparison of common colour edge detectors is available in [82].

Arguably, Canny edge detector still remains the popular and it is optimal for many types of edges [24]. However, in recent times, the wavelet-based approaches are becoming more popular because of the ability to acquire gradient images at multiple scales. Compared to Canny operator, wavelet analysis allows the combination of smoothing and differentiation operation into one computational process making it flexible and effective.

Discrete wavelet transform

The Fourier Transform is probably the most popular transform in signal processing and the case is the same in image processing where it is being used for example in frequency domain filtering, homomorphic filtering etc. The Fourier Transform decomposes the signal using basis functions that comprises of sines and cosines. Thus, a function can be decomposed into a combination of sines and cosines of varying amplitudes and frequencies.

Another transform that has become increasingly popular is called the wavelet transform [24]. Wavelets transform are particularly useful because they allow the simultaneous representation of a signal in both frequency and time domains. This is because, unlike the Fourier transform, the basis functions in wavelet transform have compact support. They are

small waves varying in frequency and duration [83]. They are obtained by translation and dilation of a scaling function *φ(x)* and a wavelet function *ψ(x)*. The translation and scaling can be generalized as:

$$\psi_{(a,b)}(x) = a^{-\frac{1}{2}}\psi(\frac{x-b}{a}) \quad 5.10$$

For convenience, it is possible to select $a = 2^{-j}$ and $b = k \cdot 2^{-j}$ to form a special class of wavelets called dyadic wavelets.

$$\psi_{(j,k)}(x) = 2^{\frac{j}{2}}\psi(2^j x - k) \quad 5.11$$

$$\varphi_{(j,k)}(x) = 2^{\frac{j}{2}}\varphi(2^j x - k)$$

Dilation of the functions is controlled by the *j* parameter while *k* controls translation. If a space V_0 is defined as the space spanned by the scaling function when *j* is zero, then $\varphi_{(j_0,k)}(x) = \varphi(x-k)$. As *j* increases, the scaling function becomes narrower representing finer resolution and as *j* decreases, the scaling function becomes wider with larger translation steps representing coarser resolution in any arbitrary space V_j. The scaling function is such that each coarser resolution subspace is a subset of the adjacent finer resolution subspace. This implies [83] that a subspace V_j can be expressed as a weighted sum of the scaling function of adjacent subspace V_{j+1}:

$$\varphi(x) = \sum_n h_\varphi \sqrt{2}\varphi(2x-n) \quad 5.12$$

The wavelet function on the other hand spans the difference between two adjacent scales V_j and V_{j+1} and is orthogonal to the scaling function. This means that a wavelet function in a subspace W_j orthogonal to V_j spanning the difference between V_j and V_{j+1} can also be expressed as a weighted sum of the scaling function of V_{j+1}:

$$\psi(x) = \sum_n h_\psi \sqrt{2}\varphi(2x-n) \quad 5.13$$

where h_φ are the scaling function coefficients and h_ψ are the wavelet function coefficients, both of which are a quadrature mirror of the other.

$$h_\psi(n) = (-1)^n h_\varphi(1-n) \quad 5.14$$

The scaling function coefficients h_φ are essentially the coefficients of a low-pass filter while h_ψ are equivalent to coefficients of a high-pass filter.

Wavelet analysis enables the decomposition of a signal into basis functions up to any desired level with each level corresponding to coarser resolution or lower frequency band [84]. In the discrete domain, a function can be represented as:

$$f(x) = \sum_{k} W_{\varphi}(j_0, k)\varphi_{(j_0,k)}(x) + \sum_{j=j_0}^{\infty}\sum_{k} W_{\psi}(j, k)\psi_{(j,k)}(x) \quad 5.15$$

Equation 5.15 is the inverse discrete wavelet transform with $W_{\varphi}(j_0, k)$ called the approximation coefficient and $W_{\psi}(j, k)$ called the detailed coefficients. The complementary forward discrete wavelet transform (DWT) is:

$$W_{\psi}(j, k) = \sum_{x} f(x)\psi_{(j,k)}(x) \quad 5.16$$

$$W_{\varphi}(j_0, k) = \sum_{x} f(x)\varphi_{(j_0,k)}(x)$$

j_0 is an arbitrary scale with the coarsest resolution. The DWT can be applied using a bank of filter; Figure 5.7 shows a filter bank for a three-scale DWT of a one-dimensional signal. DWT is extended to two-dimensional signals by first applying both wavelet and scaling filter to the rows and then to the columns.

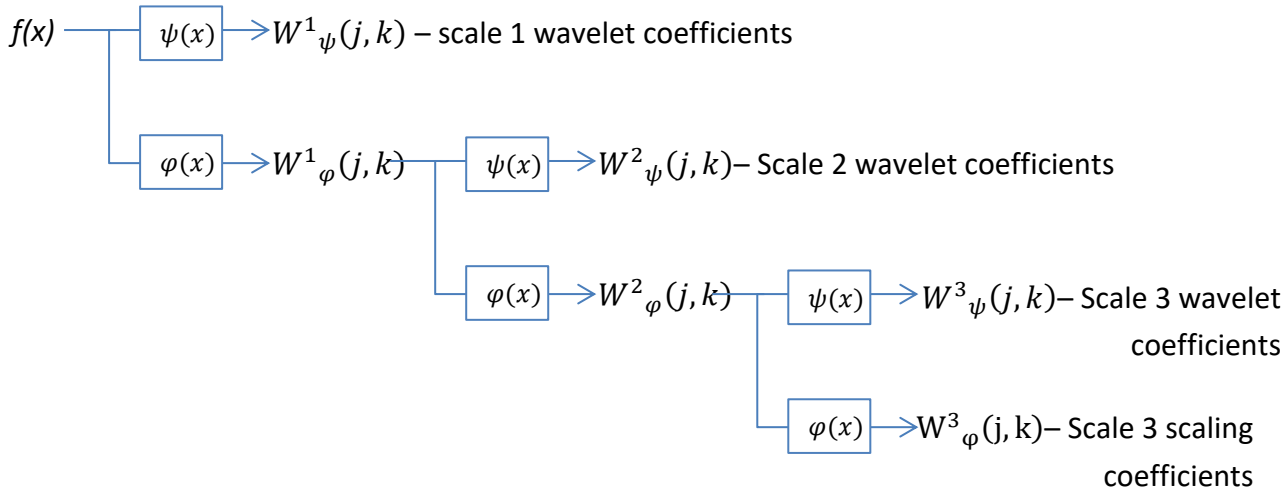


Figure 5.7: Filter bank for one-dimensional DWT

Wavelet Edge Detection

The wavelet transform has several properties that make it attractive for applications such as multiresolution analysis, clustering and localisation [85][86]. It also has several applications in sub-band coding, image denoising, image fusion, image compression and edge detection. Many wavelet-based approaches to edge detection exists in literature. In fact, Mallat and Zhong [87] implemented the popular canny edge detector using DWT.

This was achieved by constructing a dyadic wavelet which is an approximation of the first derivative of the Gaussian [88]. They also proposed the computationally efficient Fast DWT

whereby, the signal is downsampled after every level of transformation. Every other row and column is decimated so that the new signal is a quarter of the size at this previous level; this is equivalent to scaling the wavelet function by two. Another wavelet edge detection approach based on Fast DWT was implemented in [89] by decomposing the signal to a desirable level and then discard all approximation coefficients in the inverse DWT (i.e. image reconstruction) stage to derive an edge image. In [90] Canny's criterion was adopted as a guide to derive an optimal wavelet basis for edge detection.

While the Fast DWT is computationally efficient it has the disadvantage of not being shift invariant and the decimation process also has a negative impact on the continuity of spatial features that have no horizontal or vertical component [84]. In [91] a translation-invariant wavelet transform was proposed which the author reported to be similar to non-decimated (or undecimated) transform when used with the right wavelet.

The undecimated DWT (UDWT) is a variation of wavelet transform where signals are not decimated after each level of transform, instead, the scaling and wavelet filters are upsampled by inserting zeros between coefficients. Unlike the Fast DWT, the undecimated DWT is shift-invariant [84], [86], [92]. A few multiresolution edge detection techniques based on undecimated DWT have been proposed in [86], [88], [93].

Approach to Edge Detection

In this thesis, an optimal wavelet by Hsieh et al [90] was adopted for undecimated wavelet analysis technique to edge detection. This is because, wavelet analysis allows the combination of smoothing and differentiation operation into one computational process while spatial features a remain unshifted by the undecimated approach.

Figure 5.8 shows the UDWT of the dark channel of a sea image to a level of 2. Figure 5.9 shows the edge map obtained from both resolutions scale space one, two and the difference between them. In both scales, edge magnitude and direction are obtained from the vertical and horizontal detail coefficients. The edge magnitude is refined using non-maximal suppression and hysteresis thresholding. The low threshold value is chosen as 0.25 of the high threshold value while the high threshold value is set to 20.

Figure 5.8: Left to Right: DWT at level 1 and level 2 of image in Figure 5.2

The results show better performance at coarser scale. Hence, the coarser scale 2 is adopted for edge detection. However, in situations when it is really foggy or noisy, e.g. in cases where dehazing might be needed, some edges are lost in the coarser scale so a scale multiplication approach to edge detection is adopted [88]. This has the benefit of reducing noise while enhancing edges when two adjacent scales of wavelet coefficients are multiplied.

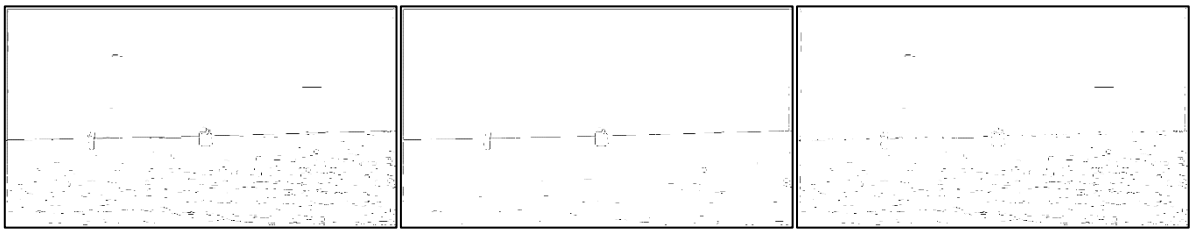


Figure 5.9: Left to Right: Edge map at scale space one; edge map at scale space two; and difference between them of image in Figure 5.8

This idea emanates from the inherent behaviour of signals across scales in the wavelet domain. The DWT amplitude will increase or remain unchanged for edges while the amplitude will decrease rapidly across scale for white noise [88]. Hence, edges will be amplified while noise is suppressed in the multiplication domain P_j.

$$P_j = W_\psi(j,k) \cdot W_\psi(j+1,k) \tag{5.17}$$

Edge points should have the same sign, so all $P_j < 0$ can be set to zero. For a two-dimensional signal two product functions are obtained:

$$P_j{}^H = W_\psi^H(j,k) \cdot W_\psi^H(j+1,k) \tag{5.18}$$

$$P_j{}^V = W_\psi^V(j,k) \cdot W_\psi^V(j+1,k)$$

W_ψ^V and W_ψ^V are the vertical and horizontal wavelets coefficients from the wavelet transform. Modulus and argument of the edge image in the multiplication domain can be obtained as:

$$M = \sqrt{P_j{}^H + P_j{}^V} \tag{5.19}$$

$$A = \tan^{-1}\left(\frac{sgn(W_\psi^V(j,k)) \cdot \sqrt{P_j{}^V}}{sgn(W_\psi^H(j,k)) \cdot \sqrt{P_j{}^H}}\right)$$

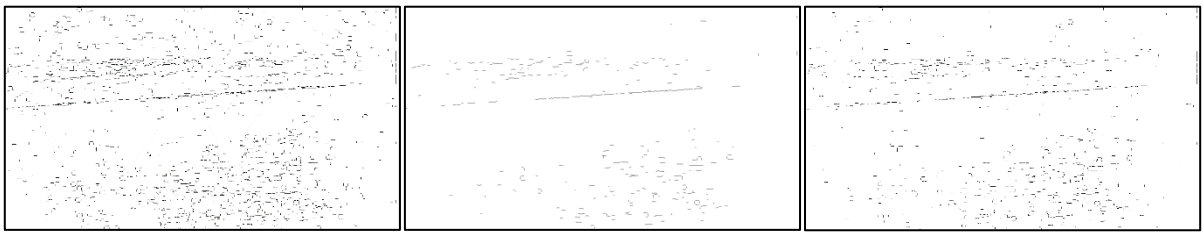


Figure 5.10: Left to Right: edge map at scale 1; edge map at scale 2; edge map in the multiplication space (of image in Figure 5.5)

Figure 5.10 shows from left to right the edge map retrieved from the finest resolution scale 1, the next coarser resolution scale 2 and the multiplication of adjacent scale 1 and 2. The result shows that some of the edges lost in scale 2 can be retrieved in the multiplication domain while noise in scale 1 is suppressed. The edge magnitude is refined using non-maximal suppression and hysteresis is used for thresholding. For this image, the high threshold value was manually set to 10 while the low threshold value is chosen as 0.25 of the high threshold value.

Automated Hysteresis Thresholding.

The objective of edge detection is to obtain a binary image for extracting features of interest; the horizon in this case. In this thesis, a hysteresis method of thresholding first introduced by Canny [94] is adopted. To make the system fully automated, there is need to implement an automated thresholding algorithm. However, automatic determination of parameters for hysteresis thresholding is not a trivial problem. A few method have been suggested by Medina et al [95]–[97]. Their methods adopt an iterative approach which can be computationally expensive.

However, a number of automated thresholding algorithms have been developed over the years that are computationally less expensive and some of the most popular methods include ones based on maximising between class variance [98], entropy of the input image [99] and preserving of some of the moment in the input image [100]. There is on-going work in this area and the reason is because no one method is optimal in all cases [101]; for example, while these methods are sufficient for segmenting multi-modal images, they are not well suited for unimodal images such as low-magnitude edge images. A technique specifically designed for unimodal images has been proposed; the so called triangular method [101].

An approach to automatic determination of hysteresis parameters commonly used in literature is to first find the high threshold t_{high} and then obtain t_{low} from the ratio $t_{high}/t_{low} = r$, where r is a constant greater than or equal to 1. This is based on the observation that the frequency of edge pixels is smaller than non-edge pixels [102]. This was the approach adopted in this thesis for simplicity and computational efficiency.

A combination of the classical optimal global thresholding technique by Otsu [98] and a more recent technique by Rosin [101] are employed for finding t_{high}. While Otsu method is optimum for multimodal edge-images, Rosin's method is more appropriate for unimodal low-magnitude ones [103]. For example, the edge image can be multimodal for a clear image taken on a bright sunny day and unimodal in hazy images.

In the system developed, t_{otsu} is first obtained using Otsu's method. The separability of the classes s, is estimated and then t_{rosin} is obtained from Rosin's method. The final high threshold value is obtained as:

$$t_{high} = \begin{cases} t_{otsu} & s > t_s \\ t_{rosin} & s < t_r \\ (2t_{rosin} + t_{otsu})/3 & otherwise \end{cases} \quad 5.20$$

The parameter t_s is selected as the value of s above which t_{otsu} is reliable and t_r as the value below which t_{otsu} is completely unreliable.

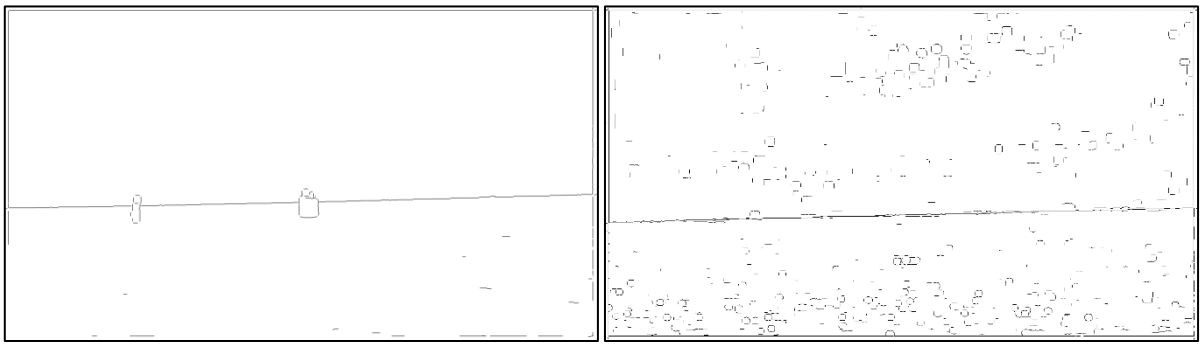


Figure 5.11: Result of edge map obtained using the automated thresholding technique presented. Left: image in Figure 5.2 and Right: image in Figure 5.3

Images in Figure 5.11 show that the system works well. However, in some cases, the system fails because it is relying solely on the separability parameter to decide the modality of the edge-image. It is possible for Otsu's method to give a poor threshold value while still estimating a separability greater than t_s. This is the case in some unimodal edge images where threshold is underestimated by Otsu's method. This is because of the very large peaks in the lower end of the histogram of such images. For equation 5.20 to be valid there is need to specify another parameter equivalent to the modality of the edge image adding another layer of complexity. However, experiments show that t_{rosin} tends to be greater than t_{otsu} in these cases thus:

$$t_{high} = t_{rosin} \quad for \;\; s > t_r \; and \; t_{rosin} > t_{otsu} \quad 5.21$$

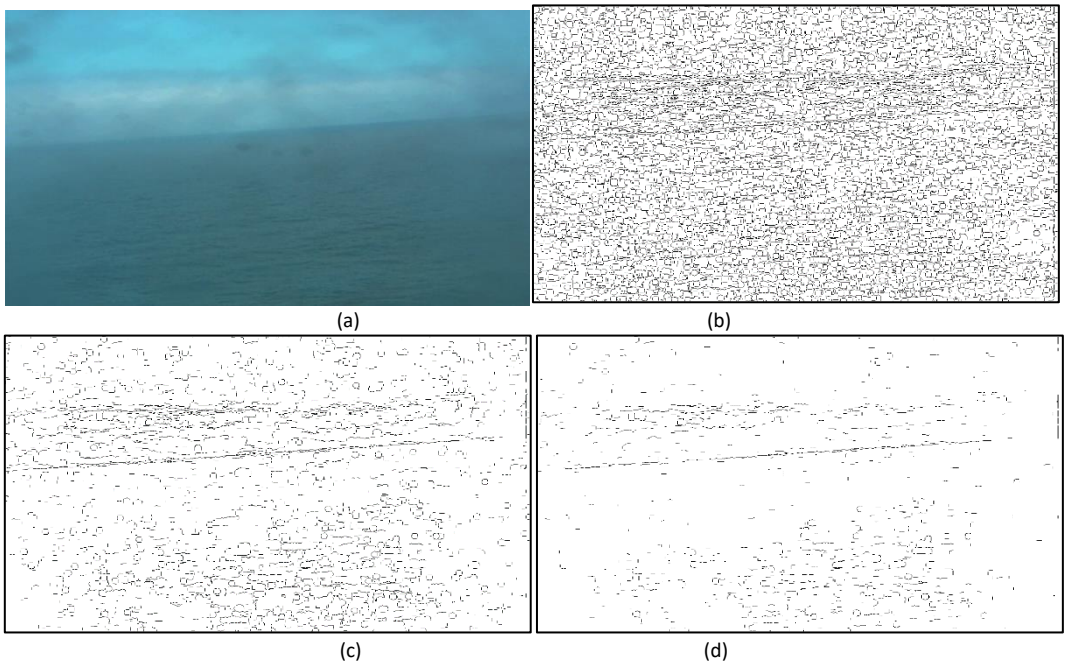


Figure 5.12: (a) original image which is a dehazed version of image in Figure 5.5 (b) edge map with threshold under estimated using Otsu (c) improved edge map (d) edge map achieved by manual thresholding.

Figure 5.12 (b) and (c) shows the result of automated hysteresis of the edge image of (a) without and with the condition in equation 5.21 respectively. Figure 5.12 (d) shows the result achieved compared to manual threshold. As can be seen in the original image, a veil of dirt in the camera lens makes it particularly difficult for edge detection.

Automated undecimated Wavelet-based Edge Detection.

So, an edged detection algorithm based on wavelet transform has been developed. The input to the algorithm is the dark channel image and the output is a binary edge image needed for horizon detection. To summarize the algorithm;

- In clear weather, the image is transformed to the wavelet domain up to level 2. The edge magnitude is then calculated from the horizontal and vertical detail coefficient at resolution scale 2.
- In foggy weather (i.e. when dehazing) is needed, the image is decomposed using wavelet transform to level 2 as before. Then, adjacent scale multiplication is used to obtain the vertical and horizontal components of the edge image.
- After edge detection, the image is further processed by applying non-maximal suppression and hysteresis respectively. The low and high threshold value used for hysteresis is obtained using the automated hysteresis algorithm proposed.

5.1.3 Horizon detection

The most common method of finding a line in literature involves using mathematical equation that describes the line in a technique called the Hough transform. The Hough transform is an evidence gathering approach. It works by mapping edge points in the image space to a parametric space defined by the equation of a line:

$$\rho = x\cos\theta + y\sin\theta \qquad 5.22$$

The idea is that, each point (x_i, y_i) on a line in the image space transforms into the same sinusoid (ρ, ϑ) in the parametric space. Therefore, by gathering evidence of (ρ, ϑ) in the parametric space, lines in an image can be found. This technique is particularly popular because of its robustness to noise and occlusion. However, the fact that every edge point in the image space is considered in the transform process means it is relatively slow, hence the need for a quicker alternative. Fast Hough transform methods have also been proposed, however they often require edge direction information. Accurate estimation of edge direction is not trivial and is often very difficult in noisy images.

As mentioned in earlier section, Wavelet analysis is particularly useful for many computer vision applications including singularity detection. Qi et al have proposed a very simple technique for finding the horizon line based on wavelet transform [68]. The approach involves constructing a one-dimensional vector from a binary edge image consisting of the numbers of edges in each row. By analysing the wavelet coefficient of the 1D vector at

coarser resolution, it is possible to find the coordinates of the ends of the horizon line. These coordinates coincide with the row with the smallest coefficient v_s and the maximum coefficient v_m between rows 1 and v_s.

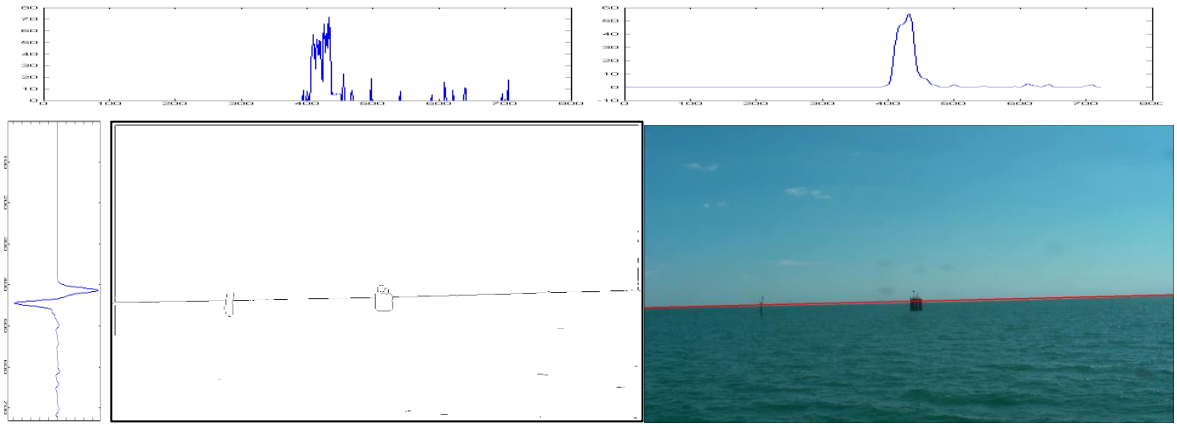


Figure 5.13: Top Left to Right: One dimensional vector of binary edge image; scaling coefficient at coarser scale 3; Bottom Left to Right: wavelet coefficient at coarser scale 3; edge map; detected horizon.

Figure 5.13 shows the plot of the number of edges per row of the edge image. The edge histogram plot is then decomposed using Hseish [90] wavelet to level 3. The scaling coefficient and the wavelet coefficient are shown. By finding v_s and v_m the left and right *y*-coordinate of the horizon can be obtained. The left coordinate refers to coordinate where $x = 0$ i.e. $(0, y_l)$ while the right coordinate is where *x* is equal to image width i.e. $(image\_width, y_r)$. Henceforth, the horizon location based on the left and right *y*-coordinates is referred to as (y_l, y_r) whereby, in the above example, $y_l = v_s$ and $y_r = v_m$.

Maximum likelihood orientation ambiguity resolution

The wavelet approach is sufficient for determining the mid-point of a line as used in [68] and coordinates of a straight line. However, an inherent drawback of this approach to horizon detection is that it results in orientation ambiguity since it is impossible to determine the orientation of the horizon line from the wavelet coefficients alone. For example, an image and its flipped version both have the same wavelet coefficient as shown in Figure 5.14. This means that any line found using this method has two possible orientations:

$$\theta_1 = \tan^{-1}\left(\frac{v_s - v_m}{image\ width}\right)$$

$$OR \qquad 5.23$$

$$\theta_2 = \tan^{-1}\left(\frac{v_m - v_s}{image\ width}\right)$$

To resolve the orientation ambiguity problem, a probabilistic approach adopted is described next.

Figure 5.14: Wavelet coefficient of an image and its flipped version is the same.

If two edge points are chosen at random, the probability that they both belong to the horizon line of orientation θ_L is given as:

$$P_{\theta L} = \left(\frac{E - N}{E}\right)^2 \tag{5.24}$$

Where *E* is the total number of edges in the binary image and *N* is the number of noisy edges. The probability that the chosen pair form a line of an orientation different to that of the horizon line, meaning that at least one of the chosen edge point is noise, can be obtained as:

$$P_{\theta N} = 1 - P_{\theta L} = \frac{2N}{E} - \left(\frac{N}{E}\right)^2 \tag{5.25}$$

Suppose the horizon line has an orientation of θ_1 and two edge points are picked at random in the region of the image between v_m and v_s, the probability that they lie on a line of orientation θ_1 will be equal to 1 in an ideal situation with no noise. Assuming the edge image is corrupted by random Gaussian noise, the probability that the chosen points lie on a line of orientation θ_1 must be greater than θ_2 (or any other angle other than θ_1) for detection to be reliable.

If the orientation of the horizon line can only take an integer value of $0 < \theta_L \leq 180$ and that the noise is identically (normally) distributed among the other (180 – 1) possible false orientation; then the probability that the chosen pair of points result in a false orientation can be obtained as:

$$P_\theta = \frac{1}{179} \times P_{\theta N} \tag{5.26}$$

From equations 5.25 and 5.26, it can be shown that the ratio of noisy edges to the total number of edges (*N/E*) must be equal to 0.94 for $P_{\theta L}$ to be equal to P_θ i.e. a probability of $P_{\theta 1} = P_{\theta 2} = 0.0036$. In this case, it means that the edge image is too noisy for detection to be reliable. So, for each pairs of points (x_1, y_1) and (x_2, y_2) picked at random between v_m and v_s, their orientation can be calculated as:

$$\theta = \tan^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \tag{5.27}$$

After Q random pair of points, $P_{\theta 1}$ and $P_{\theta 2}$ can be calculated by counting the number of pairs whose slope equalled θ_1 and θ_2 respectively. The orientation of the detected line is thus chosen as that with the greater probability. In this thesis, Q is chosen empirically as:

$$Q = \begin{cases} \max\left({E^2}/{50}, 200\right) & for\ 20 < E < 200 \\ 800 & for\ E > 200 \\ E \times (E-1) & otherwise \end{cases} \quad 5.28$$

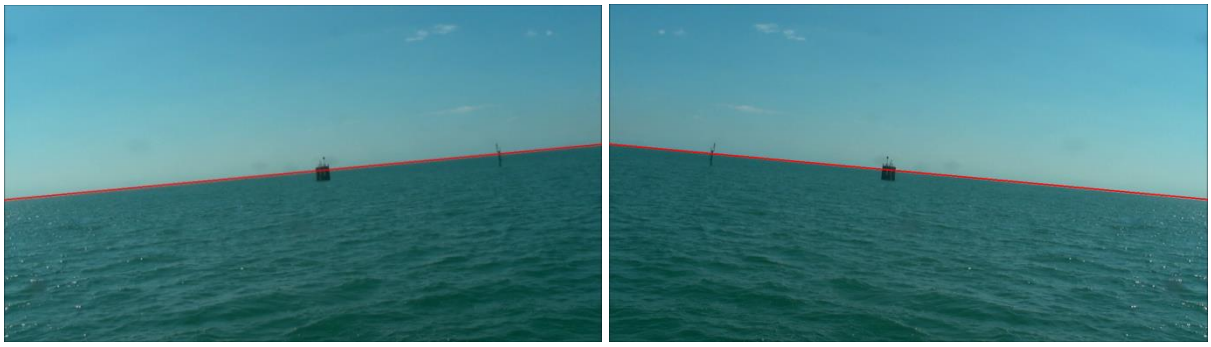


Figure 5.15: Horizon detection result after resolving orientation ambiguity

The maximum likelihood approach has the advantage of being fast and providing a check for the detection system. So, if the orientation with the greater probability is less than 0.0036, then it is obvious that there is too much noise for the detection to be reliable. In this thesis, horizon detection is said to have failed if the probability calculated is less than or equal 0.006. Figure 5.15 shows results obtained using the orientation resolution approach.

Wavelet-based Horizon Detection with orientation ambiguity resolution

Combining the wavelet singularity analysis with the maximum likelihood orientation resolution method, a fast, accurate and effective horizon detection system is obtained. The coordinates of the end of the horizon line (y_l, y_r) is obtained as, $(y_l = v_m, y_r = v_s)$ or $(y_l = v_s, yr = v_m)$ when the slope of the line is positive or negative respectively.

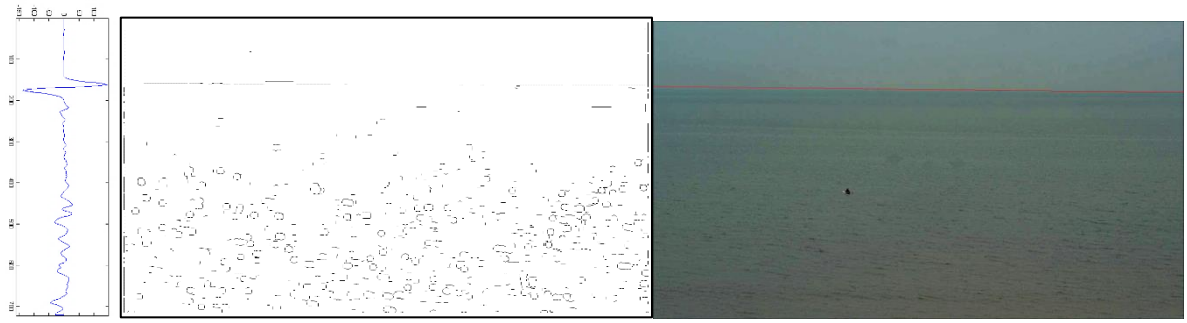


Figure 5.16: Horizon detection using presented method

As with any other Horizon detection approach, the main sources of error in this approach are due to noise, distractions and occlusion. At coarser resolution, a lot of the high frequency noise is filtered out as shown in Figure 5.16 and Figure 5.17; hence the technique works well in the presence of noise.

Figure 5.17: Additional Horizon detection result using the presented method

5.1.4 Horizon tracking

After finding the horizon, the system employs a state estimation framework for tracking it. This has the benefit of:

1. Improving speed of the system for real-time operation
2. Improving accuracy of horizon detections
3. And damping noisy (false) detections.

The Kalman filter (discussed next) and the particle filter are some of the most common state estimation frameworks that exist in literature. The Kalman filter is the most common and widely used because it is linear and optimal if noise is normally distributed or can be effectively modelled as such [52]. The particle filter is particularly suited to complex non-linear problems, thus the simpler approach based on Kalman filter is preferred here.

Kalman Filter

In most systems including this one, the state of the system is called a dynamic state since it is continually changing with time. Bayesian estimation is a probabilistic approach to estimating the state of a system from a set of noisy measurements using a mathematical model. The Kalman filter is a recursive Bayesian estimation system where the system being modelled is assumed linear and the noise is normally distributed [104]. The filter is recursive since it incorporates measurements as they arrive; this property makes it attractive for incorporation in real-time systems such as this.

Consider the state s_k of a dynamic process; a model which describes the evolution of the system with time called the transition equation can be defined as:

$$s_k^- = F s_{k-1} + w_{k-1} \tag{5.29}$$

The model is considered to contain random white noise hence the term w_{k-1}. And F is a square matrix called the transition matrix that defines the relation of the current state to the previous states. Assuming a new measurement z_k is available at time k, another model which relates the measurement to the current state can be defined as:

$$z_k = H s_k + v_k \tag{5.30}$$

The measurement is also assumed noisy hence the term v_k. The model assumes that the measurement is linearly related to the state equation [104] and *H* is a measurement matrix that relates the two of them. The process and measurement noise are assumed to be independent of each other and white, hence they have a probability distribution[105]:

$$p(w) \sim N(0, Q) \qquad 5.31$$

$$p(v) \sim N(0, R)$$

where *Q* and R are the process noise and measurement noise covariance respectively. The Kalman estimator seeks to linearly combine the measurement z_k and the priori estimate (or prediction) s_k^- to form an optimum posteriori estimate s:

$$s_k = s_k^- + K_k(z_k - Hx_k^-) \qquad 5.32$$

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

The error in the priori (e_k) and posteriori ($\tilde{e}_k$) estimates can be assumed to have error covariance P_k and $\tilde{P}_k$ respectively. The parameter K_k is called the Kalman gain or blending factor; it seeks to reduce the posteriori estimate covariance [105] and is given by equation 5.32. The Kalman filter is implemented in two steps using two groups of equations:

1. Prediction Step:

$$s_k^- = F s_{k-1} + w_{k-1}$$

$$P_k^- = F P_{k-1} F^T + Q$$

2. Correction Step:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

$$s_k = s_k^- + K_k(z_k - Hx_k^-)$$

$$P_k = (I - K_k H) P_k^-$$

The Extended Kalman filter (EKF) is a variation that extends the classical Kalman filter to non-linear processes (the reader is referred to [105] if they wish to read more). The prediction and correction equations for a process $s_k = f(s_{k-1}, w_{k-1})$ with measurement $z_k = h(s_k, v_k)$ are:

1. Prediction Step:

$$s_k^- = f(s_{k-1}, 0)$$

$$P_k^- = F_k P_{k-1} F_k^T + W_k Q_{k-1} W_k^T$$

2. Correction Step:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_{k-1} V_k^T)^{-1}$$

$$s_k = s_k^- + K_k(z_k - h(s_k^-, 0))$$

$$P_k = (I - K_k H_k) P_k^-$$

The major difference between the Kalman filter and EKF is that *H* and *F* are no longer constant but are updated at every new estimate *k*.

F_k is the Jacobian matrix of partial derivatives of *f* with respect to *s*,
W_k is the Jacobian matrix of partial derivatives of *f* with respect to *w*,
H_k is the Jacobian matrix of partial derivatives of *h* with respect to *s*, and
V_k is the Jacobian matrix of partial derivatives of *h* with respect to *v*.

Implementation of Extended Kalman Filter

To track the horizon line, the process is modelled using the polar equation of a line given as:

$$\rho = x cos\theta + y sin\theta \qquad 5.33$$

This equation is non-linear, hence an EKF is used. The state of the horizon line *(s)* is defined as the angle of rotation θ, about its centre *(x, y)* at a distance *ρ* from the origin of the image (0, 0) coordinate. It is assumed that the lines are moving at constant translation velocities *(u, v)* and rotational velocity *ω.*

$$s_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \\ \rho_{k+1} \\ u_{k+1} \\ v_{k+1} \\ \omega_{k+1} \end{bmatrix} = f(s_k, w_k) = \begin{bmatrix} x_k + u_k \\ y_k + v_k \\ \theta_k + \omega_k \\ r_k \\ u_k \\ v_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} e_x \\ e_y \\ e_\theta \\ e_\rho \\ e_u \\ e_w \\ e_\omega \end{bmatrix}_k \qquad 5.34$$

where;
$$r_k = (x_k + u_k)\cos(\theta_k + \omega_k) + (y_k + v_k)\sin(\theta_k + \omega_k)$$

For the discrete Kalman filter implementation, at time steps $\Delta t = 1$:

$$F_k = \begin{bmatrix} 1 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \Delta t \\ a & b & 0 & 0 & a & b & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_k \qquad 5.35$$

where;
$$a = \cos(\theta_k + \omega_k);\ b = \sin(\theta_k + \omega_k);$$

From the horizon detection algorithm, the left and right coordinated of the horizon line (y_l, y_r) is obtained. Hence it is easy to obtain the measurement of the centre of the line (x, y) and the orientation of the line θ. Therefore:

$$z_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ \rho_k \end{bmatrix} = h(s_k, v_k) \tag{5.36}$$

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.37}$$

Other matrices *W* and *V* are calculated as identity matrices of size 7 by 7 and 4 by 4 respectively. Since the horizon line spans from one edge of the image to the other, the centre coordinate on the *x* axis is always the same. Hence, the above model can be further simplified if x_k is substituted for a constant equal to half of the width of the image and u_k for zero.

The initial process noise covariance is set to a value of 0.01 while the initial measurement noise covariance is set to 1e-6. The measurement and process noise covariance are updated to take a value depending on the deviation (*d*) of the current measurement from previous estimates. This is because vessel movement is gradual; therefore, the movement of the horizon between successive frames is assumed to be gradual. Hence;

$$Q = \begin{cases} I \cdot 1000 & when\ d > d_T \\ I \cdot 1e-6 & otherwise \end{cases} \tag{5.38}$$

where ***I*** is an identity square matrix of appropriate dimension. The deviation of two successive measurement $z_k(y_{l1}, y_{r1})$ and $z_{k+1}(y_{l2}, y_{r2})$ can be calculated using equation 5.39. Experiments suggest that a reasonable value for d_T is 50 for image resolution 1280 by 720.

$$d = ((y_{l1} - y_{l2})^2 + (y_{r1} - y_{r2})^2)^{\frac{1}{2}} \tag{5.39}$$

Figure 5.18: Result of the horizon tracking system in various frames in a video. Top Left is the first image frame 1; the predicted horizon is initialized to the centre of the image. (Note: Red and Yellow line may be difficult to see if they overlap)

A result of the horizon tracking system is shown in Figure 5.18. The green line represents the predicted horizon location; the red line is the measured location while the yellow line is the corrected horizon location. The result shows that the adopted model can provide a good prediction of the position of the horizon just before a measurement is made. As a result, the tracker is used to improve the Horizon detection system.

Improving Horizon Detection

As mentioned earlier, the main sources of error in the horizon detection system are due to noise, distractions and occlusion. Since the tracker can give a good estimate of the next position of the horizon, the horizon detection system can be improved by limiting the next search to a desired region of interest (ROI). Predicted coordinates (y_{lp}, y_{rp}) are obtained from the priori estimate s_k^- and search is limited to a ROI of $(y_{lp} + \sigma, y_{rp} + \sigma)$ where σ is a constant.

The ROI is only set when the deviation calculated using equation 5.39 between prediction and correction are less than a specified threshold i.e. when the tracker is settled. The use of ROI does not only improve the effectiveness of the horizon detection system but also improves speed since the area of the imaged to be searched is limited. This improvement in speed is critical for achieving a reasonable real-time processing frame rate.

Another benefit of the tracker comes from the chosen value of measurement noise covariance. When the confidence of measurement is low (i.e. p = 1000), the prediction (i.e. priori estimate) is favoured more that the measurement in the posteriori estimate. This helps to dampen the effect of a brief false positive detection. As demonstrated in Figure

5.19, the corrected horizon location (i.e. the yellow line) gives the correct location of the horizon even though the measurement (i.e. the red line) gave a wrong estimate.

Figure 5.19: Result of Kalman filter dampening effect of brief false positive detection. Left: Improved horizon detection; Right: False detection due to waves on the beach. (Note: Red and Yellow line may be difficult to see if they overlap)

5.2 Multi-sensors horizon tracking system

The horizon tracking system described so far enables independent tracking of horizon in each camera that make up a multi-sensor system. However, if such system is fully calibrated, horizon tracking may not be necessary for individual sensor. Thus, once the horizon is resolved in one sensor, the detection in the other sensor effectively becomes a correspondence problem. This can speed up processing dramatically.

5.2.1 Stereo-Camera Horizon tracking

In Chapter 3, the process of estimating the rotation and translation matrices that transform a point in the image plane of one camera to the other was described. Result of the rectification performed was also shown. The rectification process greatly simplifies the stereo-correspondence problem since a point at x_1 in one image lies on a row in the other d pixels apart.

$$d = x_1 - x_2 \tag{5.40}$$

It is not difficult to show that d is inversely proportional to the distance Z of the point from the stereo-rig, i.e.

$$d = \frac{T}{Z} \times f \tag{5.41}$$

where T is the separation of the cameras in the stereo rig and f is their rectified focal length. Note that $x_1 \approx x_2$ when Z is much larger than the product $(T \times f)$ and this is assumed as the case for a point located on the horizon. For example, for a stereo-rig with 0.5m camera separation located at 15m height, the distance to the horizon is roughly 13,824m; thus, a camera with a pixel focal length of 27648pixels is required to achieve a disparity (d) of 1pixel.

To summarize, once the horizon position in localized in one image, the image pair is then stereo-rectified so that the horizon position is the same in both images. The second image may then be unrectified to achieve the real horizon position. However, it is not so simply since the stereo-calibration/rectification process may have some small errors. But, this can simplify the horizon search algorithm significantly thereby saving resources.

This method is particularly useful for detecting the horizon in the thermal image because the horizon is not always well defined in that spectrum. Reflection from clouds as well as other atmospheric anomalies sometime makes the horizon somewhat blurred in the thermal image.

5.2.2 Inertial aided horizon tracking

As shown by equations in 4.8 to 4.13, the horizon position can be used to obtain the orientation of the camera in the real world. Rearranging equation 4.12, we have that $y_2 = y_1 - D$ where *D* is given by:

$$D = \frac{f_y \cdot \tan\phi}{f_x}(x_1 - x_2) = (y_1 - y_2) \tag{5.42}$$

Substituting *D* in equation 4.13 and rearranging:

$$y_1 = c_y + \frac{D(x_1 - c_x)}{(x_1 - x_2)} + \frac{-\tan\theta\left[f_y \cos\phi\,(x_1 - x_2) + f_x \sin\phi\,(y_1 - y_2)\right]}{(x_1 - x_2)} \tag{5.43}$$

The orientation of the camera can also be estimated using an inertial sensor rigidly attached to its body. If the relative orientation of the IMU to the camera is known and any temporal offsets have been removed, the camera rotation can be easily estimated from the IMU measurements. Techniques for estimating these parameters were presented in section 3.3.2 and 3.3.3.

However, it is well known in literature that the IMU can suffer from drift noise and become unreliable over time; as shown in IMU analysis presented in section 5.3.2. IMUs that have the accuracy required for unaided distance estimation are orders of magnitude more expensive. Here, a relatively cheap IMU is employed and strategies for dealing with the noise is investigated since it can provide invaluable information that can improve the performance and speed of the HoT system.

The subject of keeping good attitude measurement with noisy data from the IMU has been well investigated in literature. The most common technique involves using a Kalman filter to track IMU bias and drift. Here, an off the shelf IMU was employed which consists of an onboard Kalman Filter as well as a robust factory calibration to estimation gyro, accelerometer and magnetometer misalignments.

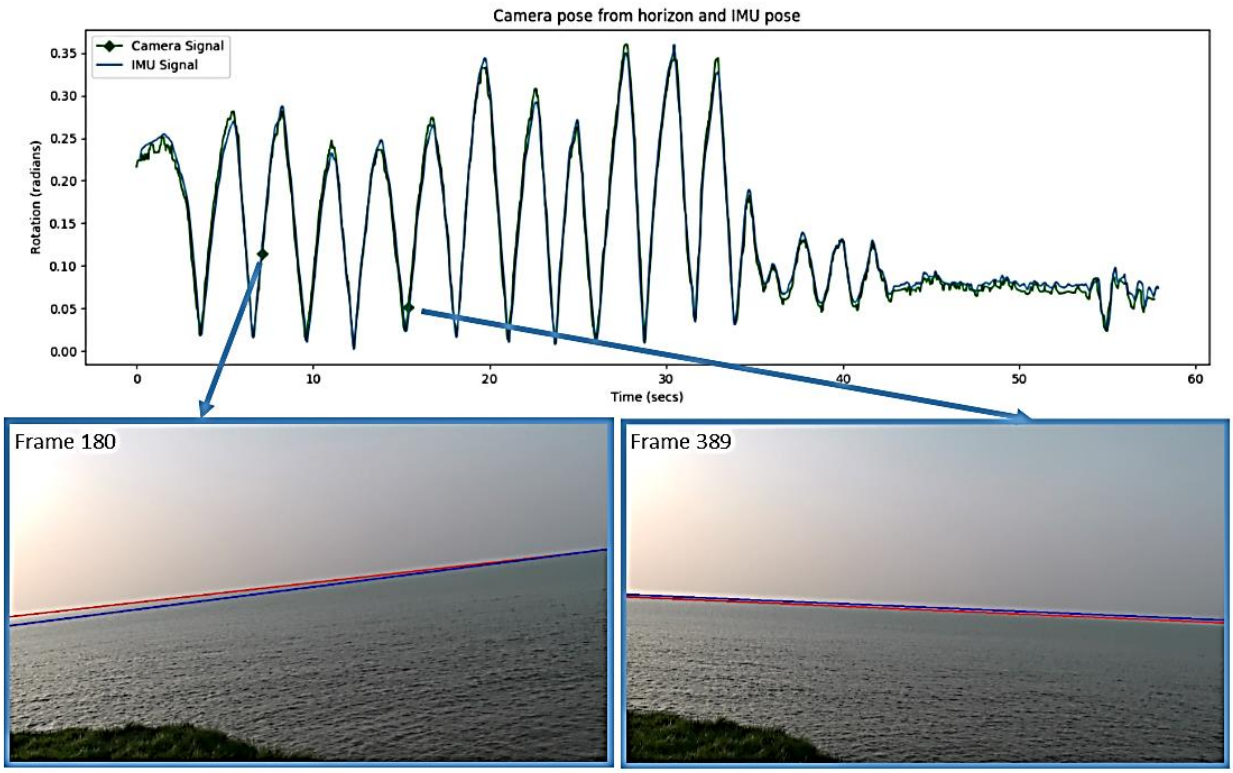


Figure 5.20: Result of horizon position obtained from image (red) compared to that estimated from IMU (blue).

Let x_1 equal to zero and x_2 equal the width of the image from the camera; given angle $\tilde{\phi}$ and $\tilde{\theta}$ obtained by transforming IMU measurements to camera coordinates, the coordinates y_1 and y_2 of the horizon in the image can be easily recovered from equations 5.42 and 5.43 as shown in Figure 5.20. Note the accent mark, this represents the fact that the camera rotation is estimated from the IMU and may contain some errors due to uncertainties caused e.g. by noise, IMU drift etc. The degree of uncertainty will depend on the quality of the IMU used and the accuracy of the calibration algorithm. These angles can be estimated directly from the image sequence albeit with uncertainty of its own, however, the optimal orientation of the camera can be estimated by fusing multiple noisy measurements. For example, this could be useful for filtering high frequency jitters in the camera measurements.

Complementary Filter

Assuming the onboard Kalman Filter in the IMU has removed any bias and drift from the IMU measurement, the task reduces to a fusion of multiple noisy measurements. The complementary filter can be very useful in this situations especially if the spectral characteristics of both sensors are complementary [106]; for example, if there are two measurements and the noise in one measurement is of high frequency and the other is of lower frequency. The general form of a complementary filter is:

$$\tilde{X} = w_1 \, Y_1 + (1 - w_1) \, Y_2 \qquad 5.44$$

Where $\tilde{X}$ is the estimate and Y_1 and Y_2 are two noisy measurements of the same signal with transfer function w_1 and $1 - w_1$ respectively for $0 \geq w_1 \leq 1$. This can be viewed as a weighted averaging function. However, rotation matrices and their quaternion equivalent have constraints that makes averaging them not straightforward. It is well known that a quaternion representing rotation must have a unit length, this property may not be preserved in the above equation. Also, a quaternion and its negative represent the same rotation, it is obvious to see why numerical averaging is not so straight forward.

A simple technique for averaging quaternion was presented in [107] which is adopted here. For a given set of quaternions $\boldsymbol{q}_i$ for $i = \{0, 1 \ldots n\}$ let $M = \sum_{i=0}^{n} w_i \boldsymbol{q}_i \boldsymbol{q}_i{}^T$. The average quaternion is given as the eigen vector corresponding to the maximum eigen value of *M.* In the case of two quaternions, a closed form solution is given as [107]:

$$\tilde{\boldsymbol{q}}_{\boldsymbol{k}} = \pm \left[\sqrt{\frac{w_1(w_1 - w_2 + z)}{z\,(w_1 + w_2 + z)}}\, \boldsymbol{q}_k^- + sign(\boldsymbol{q}_k^T \boldsymbol{q}_k^-) \sqrt{\frac{w_2(w_2 - w_1 + z)}{z\,(w_1 + w_2 + z)}}\, \boldsymbol{q}_k \right] \quad 5.45$$

where

$$z = \sqrt{(w_1 + w_2)^2 + 4 w_1 w_2\, (\boldsymbol{q}_k^T \boldsymbol{q}_k^-)^2}$$

$\tilde{\boldsymbol{q}}_{\boldsymbol{k}}$ is the optimal estimate given the new camera measurement $\boldsymbol{q}_k$ and priori estimate $\boldsymbol{q}_k^-$. For a stationary system, $\boldsymbol{q}_k^-$ is equal to the estimate at time *k-1* i.e. $\tilde{\boldsymbol{q}}_{\boldsymbol{k-1}}$. However, for a moving platform, the quaternion kinematic equation must be solved given the transition matrix $\Phi(\omega)$ related to the angular velocity ω measured by the IMU i.e.:

$$\boldsymbol{q}_k^- = \Phi(\omega)\, \tilde{\boldsymbol{q}}_{\boldsymbol{k-1}}. \quad 5.46$$

Kalman Filter

Following the review in section 2.3.4, the Kalman filter is considered here for data fusion. The state model is a non-linear system based on quaternion kinematics. Quaternion is preferred to rotation matrices because there are no singularities in the quaternion space. A comprehensive survey of non-linear filtering methods for attitude estimation can be found in [108]. They stated that despite the promise of new techniques, estimators based on Kalman filter remain sufficient for the majority of applications because they are well studied, well researched and very robust.

In the realms of Extended Kalman Filtering (EKF) for quaternion filtering, there are two main approaches depending on the model adopted i.e. 1) Multiplicative EKF and 2) Additive EKF. The main difference is that, while the latter uses difference between the true quaternion and the quaternion estimate to define the error, the former defines the error as a small

rotation between the estimated and the true orientation of the local frame of reference [109]. According to Crassidis et al. [108], the performance of the Multiplicative EKF is comparable to that of the Additive EKF. In this thesis the Additive EKF presented in the work of Choukroun et al. [110] is preferred because it is intuitive and easy to develop. It also has an adaptive element which allows it to cope better with very noisy measurement, inaccurate model and poor initialisation [110]. Although Multiplicative EKF is often used in computer vision application, we demonstrate that additive EKF are equally applicable. The Quaternion Kalman Filter (QKF) model adopted here follows the one presented in [110] and is derived next.

Given a vector ***r*** in the global coordinate frame, the corresponding vector measured in the camera frame ***b*** is given by the relation;

$$\boldsymbol{b}_C = \boldsymbol{q}_{CG} \circledast \boldsymbol{r}_G \circledast \boldsymbol{q}^*_{CG} \qquad 5.47$$

$$\boldsymbol{b}_C \circledast \boldsymbol{q}_{CG} = \boldsymbol{q}_{CG} \circledast \boldsymbol{r}_G$$

Using the matrix representation of quaternion product, we have:

$$\lfloor \boldsymbol{b}_C \rfloor_L \, \boldsymbol{q}_{CG} = \lfloor \boldsymbol{r}_G \rfloor_R \, \boldsymbol{q}_{CG} \qquad 5.48$$

Where $$\lfloor \boldsymbol{q} \rfloor_L = q_w I + \begin{bmatrix} 0 & -q_v^T \\ q_v & \lfloor q_v \rfloor_\times \end{bmatrix} \quad and \quad \lfloor \boldsymbol{q} \rfloor_R = q_w I + \begin{bmatrix} 0 & -q_v^T \\ q_v & -\lfloor q_v \rfloor_\times \end{bmatrix}$$

$$\lfloor q_v \rfloor_\times = \begin{bmatrix} 0 & -q_z & q_y \\ q_z & 0 & -q_x \\ -q_y & q_x & 0 \end{bmatrix}$$

thus:

$$(\lfloor \boldsymbol{b}_C \rfloor_L - \lfloor \boldsymbol{r}_G \rfloor_R) \, \boldsymbol{q}_{CG} = 0 \qquad 5.49$$

The vector ***b*** is an estimate and is assumed to be the arithmetic difference of the true vector $\boldsymbol{b}^o$ and an error term $\boldsymbol{\delta b}$.

$$(\lfloor \boldsymbol{b}^o_C \rfloor_L - \lfloor \boldsymbol{r}_G \rfloor_R) \, \boldsymbol{q}_{CG} - \lfloor \boldsymbol{\delta b}_C \rfloor_L \boldsymbol{q}_{CG} = 0 \qquad 5.50$$

$$(\lfloor \boldsymbol{b}^o_C \rfloor_L - \lfloor \boldsymbol{r}_G \rfloor_R) \, \boldsymbol{q}_{CG} - \lfloor \boldsymbol{q}_{CG} \rfloor_R \boldsymbol{\delta b}_C = 0$$

This is very similar to the measurement model used in linear Kalman filter $z_k = Hx_k + v_k$. Following the definition of extended Kalman filter given in section 5.1.4, we can obtain the measurement Jacobean matrices *H* and *V* by taking the partial derivative of equation 5.50 with respect to $\boldsymbol{q}_{CG}$ (i.e. state s_k) and noise term $\boldsymbol{\delta b}_C$ respectively:

$$H = (\lfloor \boldsymbol{b}_C \rfloor_L - \lfloor \boldsymbol{r}_G \rfloor_R) \, ; \qquad z_k = 0;$$

$$v_k = -\lfloor \boldsymbol{q}_{CG} \rfloor_R \boldsymbol{\delta b}_C \quad ; \qquad V = \lfloor \boldsymbol{q}_{CG} \rfloor_R$$

The measurement covariance matrix *R* is given as the zero-mean covariance of v_k. The correction step of the QKF algorithm is thus:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_{k-1} V_k^T)^{-1}$$

$$x_k = (I - K_k H_k) x_k^-$$

$$P_k = (I - K_k H_k) P_k^-$$

Let $\Delta\theta$ represent the axis change in rotation measured by the IMU between time *k* and *k-1* in the global coordinate frame, from the theory of quaternion kinematics [56] the relation between quaternion estimate at different times steps is given as:

$$\boldsymbol{q_k} = \Phi\, \boldsymbol{q_{k-1}} \tag{5.51}$$

where $\Phi = e^{\lfloor n \rfloor_L}$ and $n = [0, \Delta\boldsymbol{\theta}/2]^T$. $\Delta\boldsymbol{\theta}$ is assumed to contain some noise $\delta\theta$ hence, the noise free term $\Phi^o = e^{\lfloor n^o \rfloor_L}$. Note that $\boldsymbol{q}_k$ is an estimate of $\boldsymbol{q}_{BG}$ at time k, the full notation (i.e. $\boldsymbol{q}_{BG_k}$) is omitted for conciseness.

$$\Delta\Phi = \Phi - \Phi^o = e^{\lfloor n \rfloor_L} - e^{\lfloor n^o \rfloor_L} \tag{5.52}$$

Following Taylor series expansion and some development assuming that the error term $\boldsymbol{\delta\theta}$ is very small, we can show that $\Delta\Phi \approx 0.5 \times \lfloor \boldsymbol{\delta\theta} \rfloor_L$ [110].

$$\boldsymbol{q}_k = \Phi_k\, \boldsymbol{q}_{k-1} - {}^1\!/_2 \lfloor \boldsymbol{q}_{k-1} \rfloor_R \boldsymbol{\delta\theta} \tag{5.53}$$

This is similar to the transition state model of the linear Kalman filter $x_k^- = Ax_{k-1} + w_{k-1}$. The process Jacobean matrices *A* and *W* are obtained by taking the partial derivative of equation 5.53 with respect to $\boldsymbol{q}_{k-1}$ (i.e. x_{k-1}) and noise term $\boldsymbol{\delta\theta}$ respectively:

$$A = e^{\lfloor n \rfloor_L} \quad ; \quad x_k^- = \boldsymbol{q}_k;$$

$$w_k = -{}^1\!/_2 \lfloor \boldsymbol{q}_{k-1} \rfloor_R \boldsymbol{\delta\theta} \; ; \quad W = -{}^1\!/_2 \lfloor \boldsymbol{q}_{k-1} \rfloor_R$$

The process covariance matrix $\boldsymbol{Q}$ is given as the zero-mean covariance of w_k. The prediction step of the QKF algorithm is thus:

$$x_k^- = Ax_{k-1}$$

$$P_k^- = AP_{k-1}A^T + W_k Q_{k-1} W_k^T$$

The complete QKF algorithm is considered formed using the prediction and correction equations presented. Further details on estimating the measurement and process noise covariance matrices is given in [110]. The posterior quaternion estimate x_k must be normalized before use. Since it is not possible to resolve yaw rotation from horizon alone, the system is assumed to be fixed in yaw axis. Hence, when using measurements from the AHRS, the yaw component must be zeroed. The prior quaternion estimate x_k^- can be used

to predict the current position of the horizon in the image and thus speed up processing while the posterior estimate is used in equations 4.8 and 4.7 for distance estimation.

The Kalman filter implemented and tested is an augmented version of the simplistic model described here to include estimation of IMU bias. This bias can be used to detect and track additional drift that is not detected onboard the IMU and improves the predictive power of the filter. The implementation also includes a more complete solution for the measurement and process noise covariance. The implementation follows exactly the algorithm in [110] with the exception that we use three vectors instead of one; each vector corresponding to the principal axis of rotation.

The angular difference between the IMU and camera measurement in Figure 5.20 is shown in the box plot in Figure 5.21. The result clearly shows the discrepancies between both sensors. In some situations, the horizon is occluded in the camera (e.g. due to large vessel pitch or roll), this is simulated by resampling the camera estimates to 500ms, 1s and 3s from the original sample interval of 40ms. The missing camera estimates are substituted by the newest available IMU estimates in the correction step of both filters. The box plot of the angular difference between the true camera estimate and the estimates from the Kalman filter, complementary filter and IMU was compared (Figure 5.21). Result shows that the Kalman filter is best able to deal with this situation due its superior predictive capability. This capability became less effective as the sample interval is increased and the Kalman filter estimates becomes no better than raw IMU estimate.

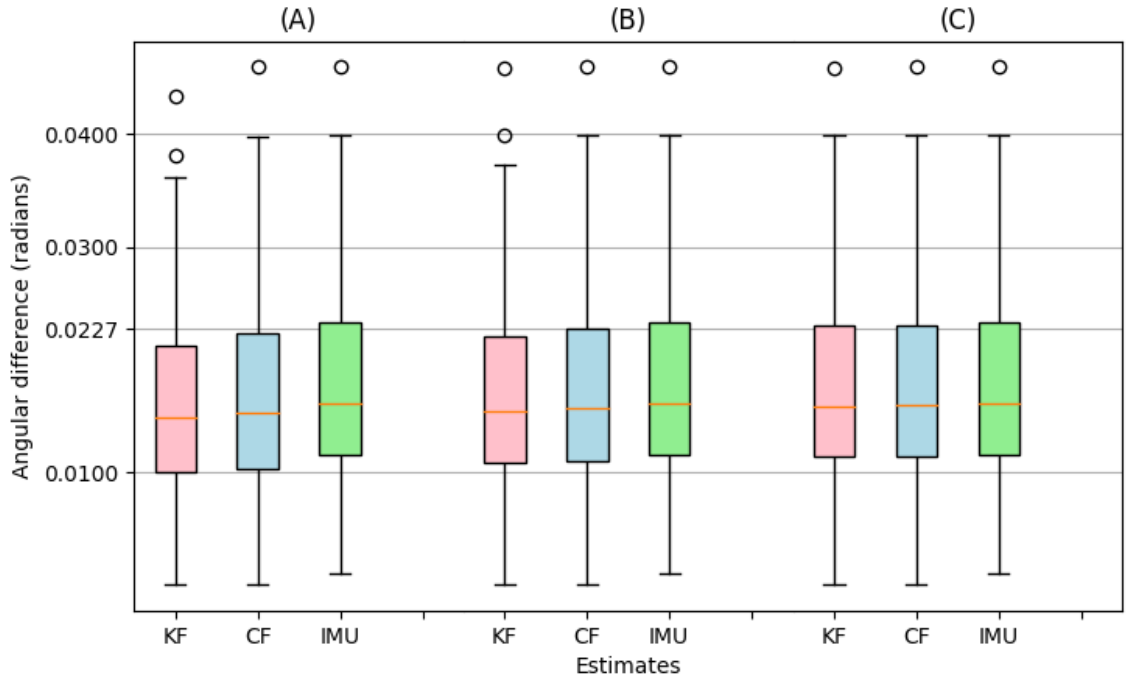


Figure 5.21: Box plot of angular difference between the camera estimate (True estimate) and estimate from Kalman Filter (KF), Complementary Filter (CF) and IMU. (A) Camera sample interval is 500ms, (B) Camera sample interval is 1s and (C) Camera sample interval is 3s.

The Kalman filter is clearly the more superior and the preferred for the HoT system, result of the filter estimate is shown in Figure 5.22. However, in the example considered here, the performance of the Kalman Filter is only slightly better than the Complementary filter because the majority of the noise is random with zero mean i.e. no IMU drift. Analysis of the bias estimated by the Kalman filter further supports this claim; the histogram plot in Figure 5.22 shows that the bias in each principal axis is very similar to a zero mean Gaussian distribution. Thus, the Complementary filter may be sufficient in situations where camera measurement is constant e.g. a non-moving platform.

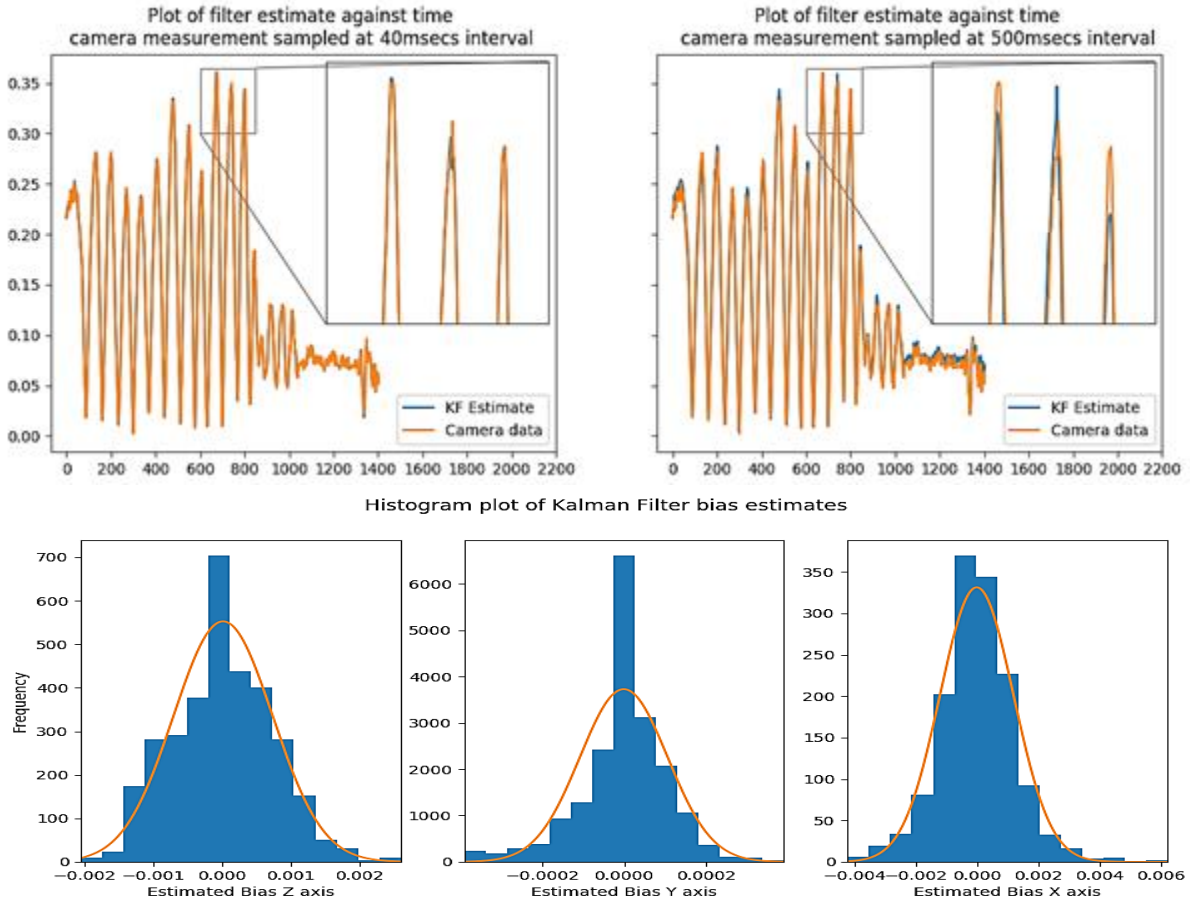


Figure 5.22: Top row: Result of Kalman filter estimates with camera estimates sampled at; Left: 40ms; Right: 500ms; Bottom Row: Histogram plot of bias estimated from Kalman filter for; Left: Z-axis; Middle: Y-axis; Right: X-axis. The bias estimate proves normality and verifies that the Kalman filter works well.

5.3 Evaluation of the HoT algorithm

An effective algorithm for detecting horizon at sea has been described in detail in the previous section. The cooperation of multiple sensor to improve the process has also been presented. The following sections present an analysis of the techniques used here.

5.3.1 Wavelet horizon detection

All the steps used in the horizon detection algorithm developed here was described in 5.1. The algorithm relies heavily on the dark channel prior in pre-processing step to achieve

consistent reliable results. Alternative approaches would involve the use of more common smoothing filters such as the Gaussian filter or perhaps, no filtering at all as in [74].

Table 5-1: Percentage of successful detection of horizon when the image is pre-processed with a Gaussian filter, No filter and Dark Channel. Result in red are below 97% detection rate.

No	Video File	Gaussian Blur 5 x 5 Kernel	No Filter	Dark channel 15 x 15 Kernel	Total Frames
1	clear	100	88.97	100	1497
2	clear	99.93	49.75	100	1443
3	foggy	46.09	20.31	97.65	256
4	glare (fixed camera)	93.84	73.84	100	130
5	glare	100	97.70	100	6105
6	gloomy	99.22	98.65	99.36	1411
7	noisy	83.42	69.71	100	175
8	very clear	97.89	91.17	97.26	476
9	very foggy (fixed camera)	100	4.93	100	709

We have conducted tests to compare these alternative approaches using nine different video data collected from multiple locations, cameras and different weather conditions; results are shown in Table 5-1. Unless otherwise specified, all the video data was collected on a moving boat i.e. horizon position was constantly changing. In cases of fixed cameras (i.e. cameras on a fixed platform), these files were tested due to the prevailing sea conditions. All the files are named in the table based on the prevailing weather conditions and some images are as shown in Figure 5.23.

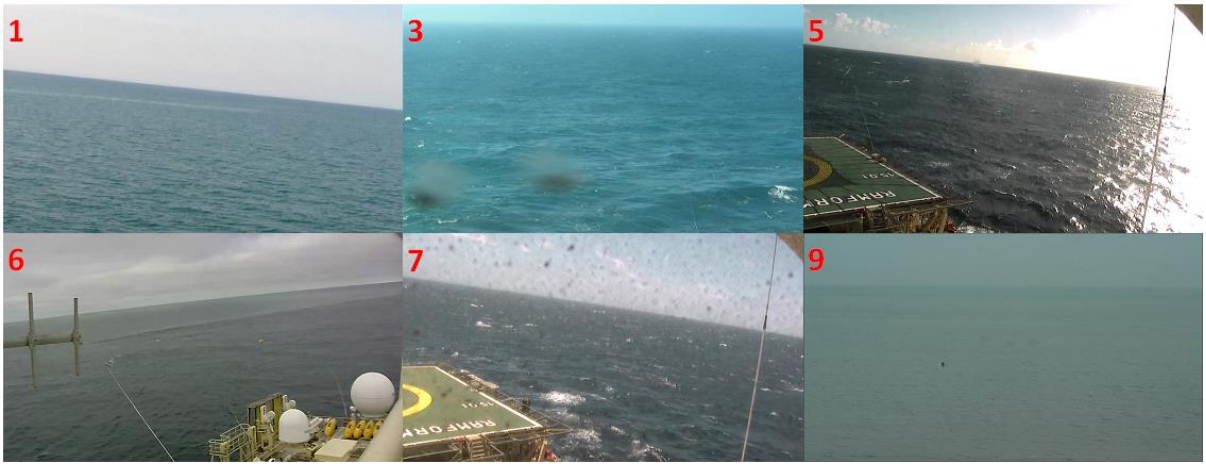


Figure 5.23: Images from video files used to test pre-processing filters, they are numbered to match the row numbers in Table 5-2. 1) clear horizon; 3) blurred horizon due to fog and droplets on camera screen 5) glare from sun masking some of the horizon; 6) gloomy images due to cloudy sky 7) noise due to dust on camera screen; 9) heavy fog.

The result shows that pre-processing with the Dark channel gave the most consistent performance across the dataset while using no filter at all, gave the worst performance. The use of a 15 x 15 kernel Gaussian filter was also tested (results not shown) but these performed worse. Using a 30 x 30 Dark channel filter gave much better result than the rest except in the case of file "very foggy (fixed camera)" where it completely failed. This failure is because no enhancement/dehazing was used in these tests. Note that these images (in the

"very foggy" video) are like those dehazed in Figure 5.4 and improved dark channel shown in Figure 5.5; thus, the preferred solution will be to dehaze the image and then use the 30 x 30 Dark channel filter.

After pre-processing (see section 5.1.1), edge detection is performed using a well establish and understood wavelet based edge detector that is similar in performance to, and in some case more effective than, the canny filter (see section 5.1.2). To localise the horizon position, a modified version of the method based on wavelet transform singularity analysis introduced in [68] has been developed (see section 5.1.3). A probability based technique has been incorporated for resolving the orientation of the detected horizon and for providing some check on the accuracy of the detection. Compared to classical Hough Transform, the proposed wavelet technique is up to 5 times faster. The average speed of the wavelet method over 1358 frames of resolution 1280 by 720 is 6millisecs while the speed of the Hough method is 30.77millisecs.

Table 5-2: Average speed of horizon detection

Test	Average speed of Wavelet method (millisecs)	Average speed of Hough Transform method (millisecs)	Resolution of frames	Total number of frames	Video sources
1	6.01	30.65	1280 by 720	1358	3
2	16.81	76.19	1706 by 1280	265	2

While the wavelet method can cope with noise due to smoothing from wavelet transform across scales, the Hough method is still the most robust. This is partially because the robustness of the wavelet method to noise reduces as the horizon line rotates from the horizontal due to the resulting spreading of horizon edges across rows. However, the wavelet method gave the most accurate localisation of the horizon position from frame-to-frame since the Hough method occasionally had slight angular errors. The angular errors may be due to the discrete nature of the accumulation space.

5.3.2 Experimental analysis of Inertial Measurement Unit

The digital camera is a self-sustaining passive sensor capable of estimating rotation with respect to the global frame by analysing reference points from frame-to-frame; in this case, the horizon is the reference point as detailed in section 4.1.2. An alternative passive sensor that may be used to estimate rotation with reference to the global frame is the IMU. An estimate of the 3-D orientation of the camera can be obtained from the yaw, pitch and roll angle measurements of IMU synchronised with the camera.

Recent improvements in performance of lightweight micro-machined electromechanical systems (MEMS) inertial sensors [111] has resulted in its increasing use in engineering applications, for example, inertial navigation system (INS). Typical IMU nowadays, consist of 3-axis accelerometer, gyroscopes and magnetometers allowing 9 degrees of freedom (DOF).

Although, the performance of MEMS inertial units has improved, their use in performance critical system is limited due to very noisy outputs from the sensors.

For example, Woodman [111] showed by experiments and simulations that for a simple INS based on Xsens IMU, the average position error grows to over 150m after 60 seconds using just gyroscope and accelerometer data. He showed that this drift is due mainly to white noise in gyroscope and accelerometer data. Fusing a magnetometer can improve this drift to around 5m after 60 seconds [111]. However, the measurements will continue to drift unless they are corrected using measurement from a more reliable sensor such as a GPS or a Camera [111]. Compared to the IMU sensor, the GPS and visual sensors have relatively lower sampling frequency, but they drift much less.

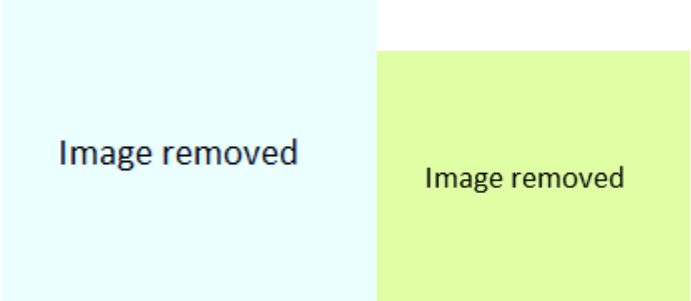


Figure 5.24: Razor 9-DOF IMU [source: https://www.sparkfun.com/products/10736]

Here, a preliminary analysis on the suitability of IMU for estimating system orientation and thus distance estimation is carried out by estimating its measurement noise and drift over time. The IMU adopted is a relatively cheap consumer grade 9-DOF device from Sparkfun called the Razor IMU. The version used for this test incorporates an ITG-3200 (MEMS triple-axis gyro), ADXL345 (triple-axis accelerometer), and HMC5883L (triple-axis magnetometer). The experiments carried out to analyse the IMU are detailed in the following sections.

5.3.2.1 Signal noise analysis

To analyse the signal noise of the sensors on the IMU, the Allan Variance technique is adopted [112]. This technique was applied to raw data output from a stationary Razor IMU collected for 7hrs; the output rate from all three sensors was 20Hz.

Table 5-3: Numerical values of noise processes from Allan deviation plot

	Bias instability	Random walk
Accel x	5.0027e-005 m/s^2	5.3849e-004 $m/s^2/\sqrt{s}$
Accel y	3.9758e-005 m/s^2	5.2158e-004 $m/s^2/\sqrt{s}$
Accel z	1.0227e-004 m/s^2	8.0927e-004 $m/s^2/\sqrt{s}$
Gyro x	2.7599e-005 rad/s	3.9128e-004 rad/s/√s
Gyro y	3.3659e-005 rad/s	0.0016 rad/s/√s
Gyro z	2.8237e-005 rad/s	4.1006e-004 rad/s/√s
Mag x	2.9483e-004 gauss	7.0038e-004 gauss/√s
Mag y	2.6930e-004 gauss	9.2904e-004 gauss/√s
Mag z	1.2994e-004 gauss	8.3804e-004 gauss/√s

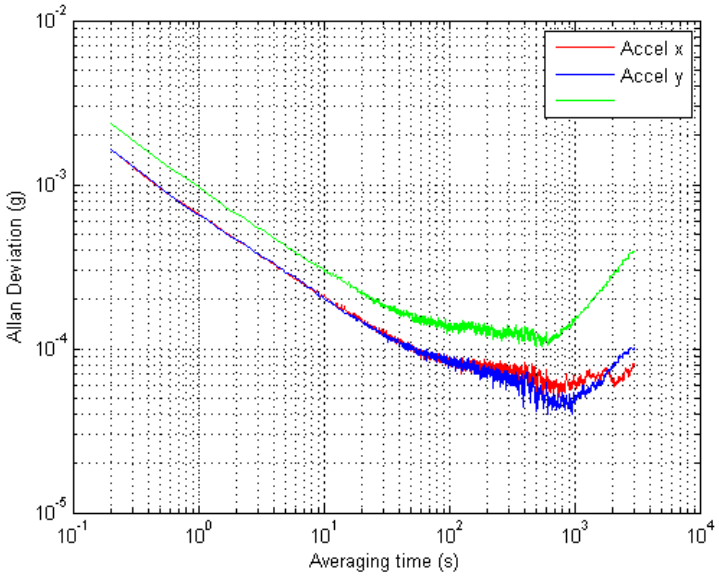


Figure 5.25: Razor IMU Allan deviation log-log plot for accelerometers

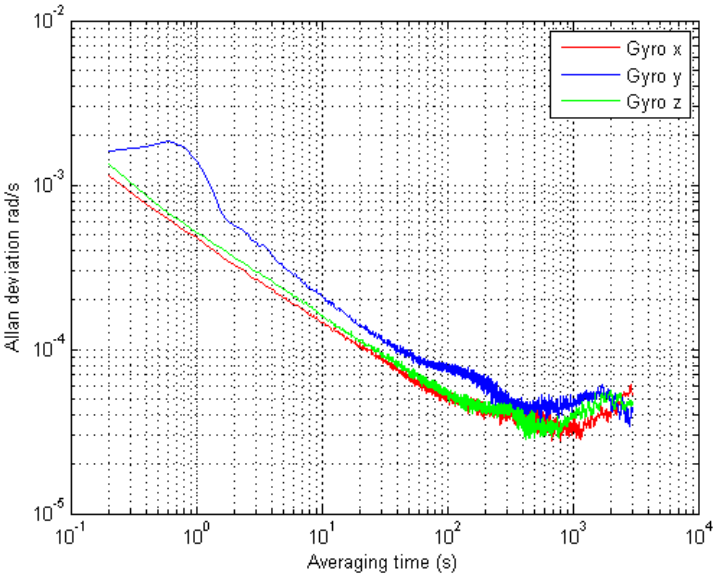


Figure 5.26: Razor IMU Allan deviation log-log plot for gyroscope

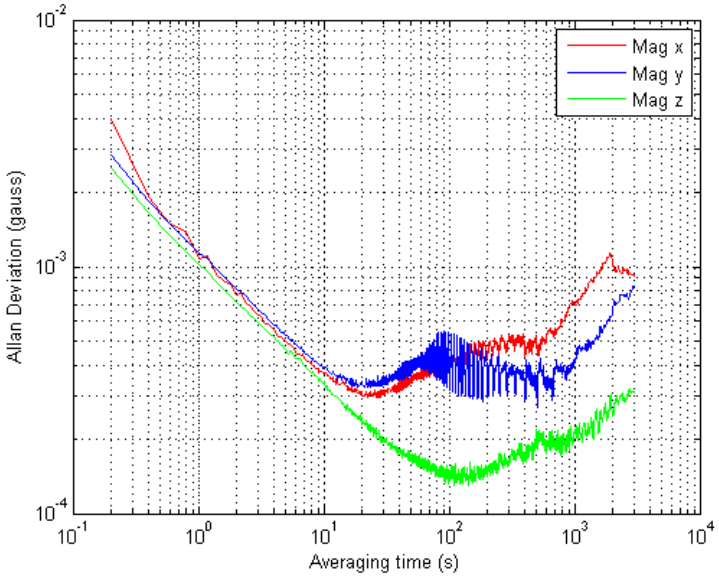


Figure 5.27: Razor IMU Allan deviation log-log plot for magnetometers

Allan variance is used to characterise the noise of the sensors on the IMU. Different process noises appear at different regions of averaging times. The numerical values of bias instability and Random walk can be obtained directly from the Allan plot [92]. Results from the Allan deviation analysis shows that the noise characteristics of the IMU sensors behaved as expected:

1. The signal is dominated by white noise (Random Walk) at short averaging times which is proportional to the square root of time.
2. Correlated noise at larger averaging times.

5.3.2.2 Attitude and Heading Reference System (AHRS) drift

In this section, the performance of the IMU as an AHRS is analysed. Data from the sensors are fused using firmware supplied by the manufacturers loaded on the microprocessor on-board Razor IMU. To test the performance, data (i.e. yaw, pitch and roll angles) is collected from the device while it is kept stationary. The IMU is calibrated beforehand to remove constant bias noises in all three sensors.

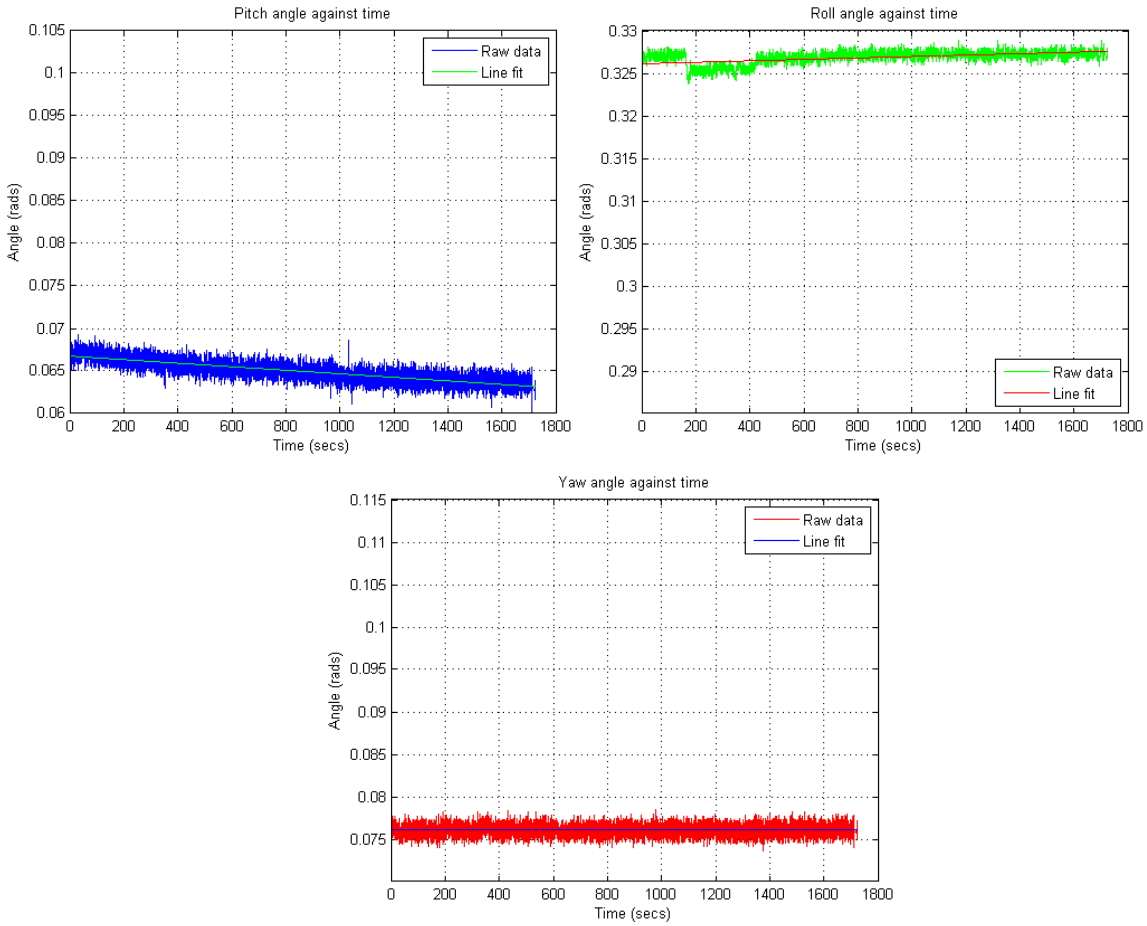


Figure 5.28: Plot of stationary IMU angle measurements with time

Figure 5.28 shows a plot of the angle measurements from the Razor IMU against time after 25mins of data collection at rate of 50Hz. A straight line is fitted to the raw data and the

slope is calculated. The result shows that, the IMU drifts with time. This operation was repeated for twelve separate data collections from the same sensor. In each case the sensor drift was completely different with no noticeable correlation. Six of the data collections were done with the IMU lying in the exact same position. Thus, it is concluded that the drift is due mainly to random walk noise in the sensors; since constant bias noise has been removed (at least to a reasonable extent) by calibration. In the example reported here (chosen at random for demonstration) it was observed that after 20mins, the absolute drift was 0.0025 rads for pitch, 0.0010 rads for roll and 2.6535e-005 rads for yaw angles.

In conclusion, the analysis on the IMU so far has shown that it is dominated by random walk noise and constant bias noise like any typical MEMS inertial sensors. Although calibration helped eliminate a significant amount of the constant noise, the IMU measurements still drifts with time. The random nature of the drift makes it particularly difficult to model and compensate. From equation 4.10, it can be shown that;

$$c_y - y + e_y = \frac{f_y}{Z} \times \frac{\left(h + Z\sin(\psi + e_\psi)\right)}{\cos(\psi + e_\psi)} \qquad 5.54$$

Figure 5.29 shows a plot of absolute error pixel position estimate e_y caused by error in pitch angle (e_ψ). Result shows that at less than 0.003° radians in pitch error, the corresponding pixel error is less than 4. This means that the drift in IMU measurement over brief period may not cause significant errors in distance estimates. This can be useful for speeding up the image processing step and/or making the system more robust. In addition, the long-term drift can be corrected using measurement from the camera i.e., an IMU may be relied upon for short term measurement and camera for long term measurements by compensating for IMU drift.

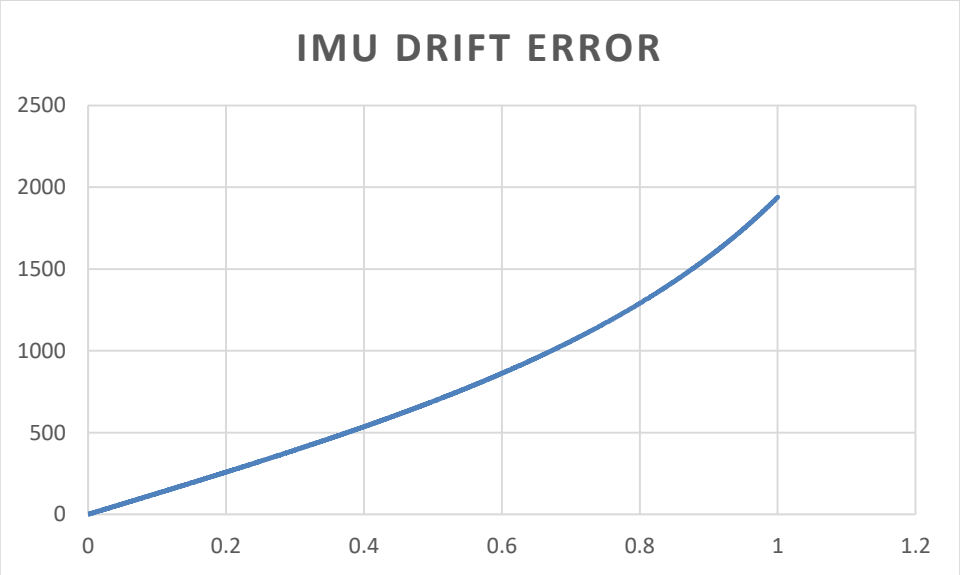


Figure 5.29: Plot of absolute error in pixel position estimate e_y against error in pitch angle (e_ψ) in radians

5.3.2.3 Analysis of Quaternion Kalman Filter

In this section, the filters designed for fusing inertial and vision measurements are evaluated using simulated and real data. Their performance and efficiency are also compared.

Simulation

Extensive Monte-Carlo simulations were performed to test the stability of the filters over an extended period. The following angular rate was used:

$$\omega(t) = [10 \quad 10 \quad 10]^T \; sin(2\pi t/10) \; \frac{deg}{s}$$

A small amount of zero mean noise with standard deviation 10 degrees and 1 degree was added to the angular rate to simulate IMU measurement and camera measurements respectively. In addition, a theoretical drift of $[1 \quad -1 \quad 0.5]^T$ degrees/hr was added to the IMU measurement. The angular error $\delta\phi$ is obtained as the magnitude of the rotation that bring the filter estimate in alignment with the true rotation. Each Monte-Carlo run lasts for a day i.e. 86,400 seconds, the mean $\delta\phi$ is plotted against time in Figure 5.30. the result shows that both filters are stable for long term operation.

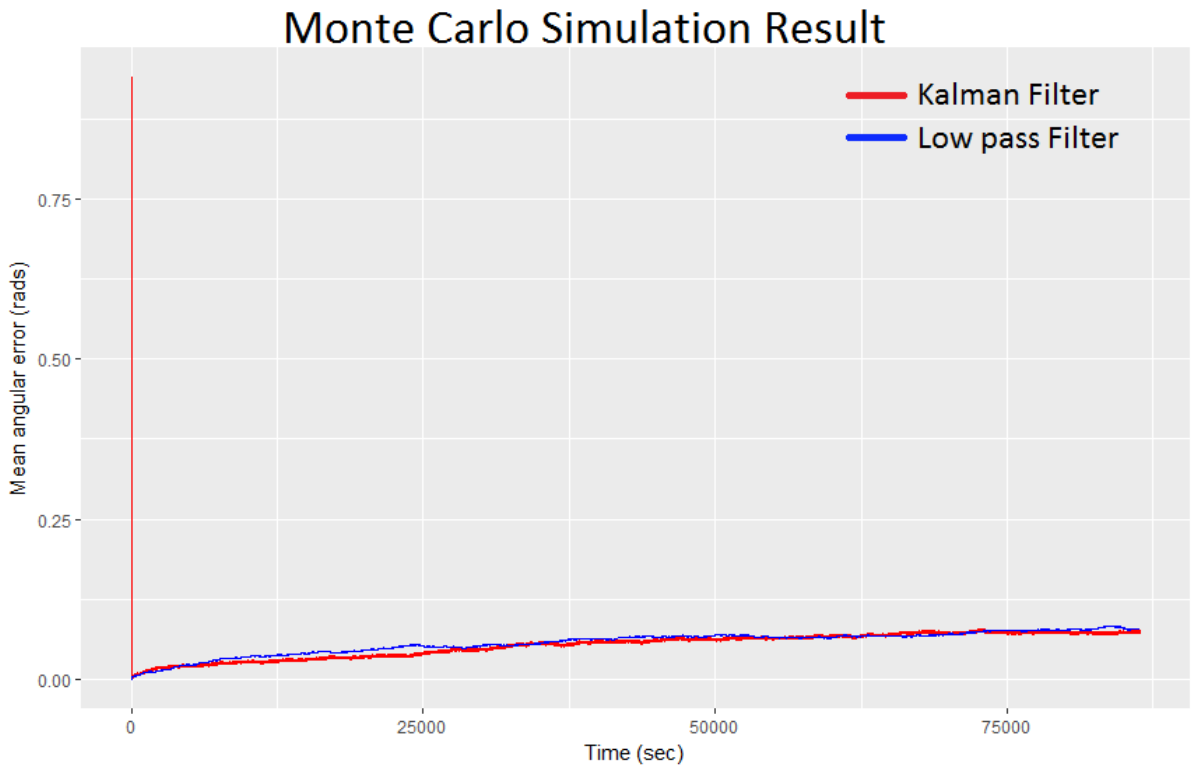


Figure 5.30: Plot of mean angular error against time from 14 times Monte Carlo simulation. Result shows that both filters are stable for long term operation.

Real data

Raw gyro and accelerometer measurement was collected from an IMU sensor kept stationary on a table for a few minutes. The IMU orientation $\boldsymbol{q_0}$ obtained from the average of accelerometer data over the entire capture duration; that is the rotation that bring the

reference gravity vector $[0 \quad 0 \quad -1]^T$ in alignment with the average accelerometer reading. At each time step k, the orientation estimate from raw gyro measurement can be obtained from equation 5.46 as $\boldsymbol{q_k} = \Phi(\omega)\, \boldsymbol{q_{k-1}}$ where $\boldsymbol{q_{k-1}}$ is equal to $\boldsymbol{q_0}$ at time k = 1. $\boldsymbol{q_k}$ is used in the predictive step of the filter while $\boldsymbol{q_0}$ is used in the correction step to obtain a filter estimate $\tilde{\boldsymbol{q}}_{\boldsymbol{k}}$ at each time step.

Again, the angular error $\delta\phi_{est}$ is obtained as the magnitude of the rotation that bring $\tilde{\boldsymbol{q}}_{\boldsymbol{k}}$ in alignment with $\boldsymbol{q_0}$. Similarly, an angular error $\delta\phi_{raw}$ is estimated for the noisy gyro measurement $\boldsymbol{q_k}$. Figure 5.31 shows the result obtained for the complementary and Kalman filter. The result shows that both filters can deal effectively with noisy measurements from the IMU. This is an exaggerated scenario since, as mentioned previously, the output of the IMU is filtered by onboard sophisticated filter. However, this proves that even in this situation, the filter can deal with noisy measurements with the added benefit that image processing is sped up with the incorporation of such measurement.

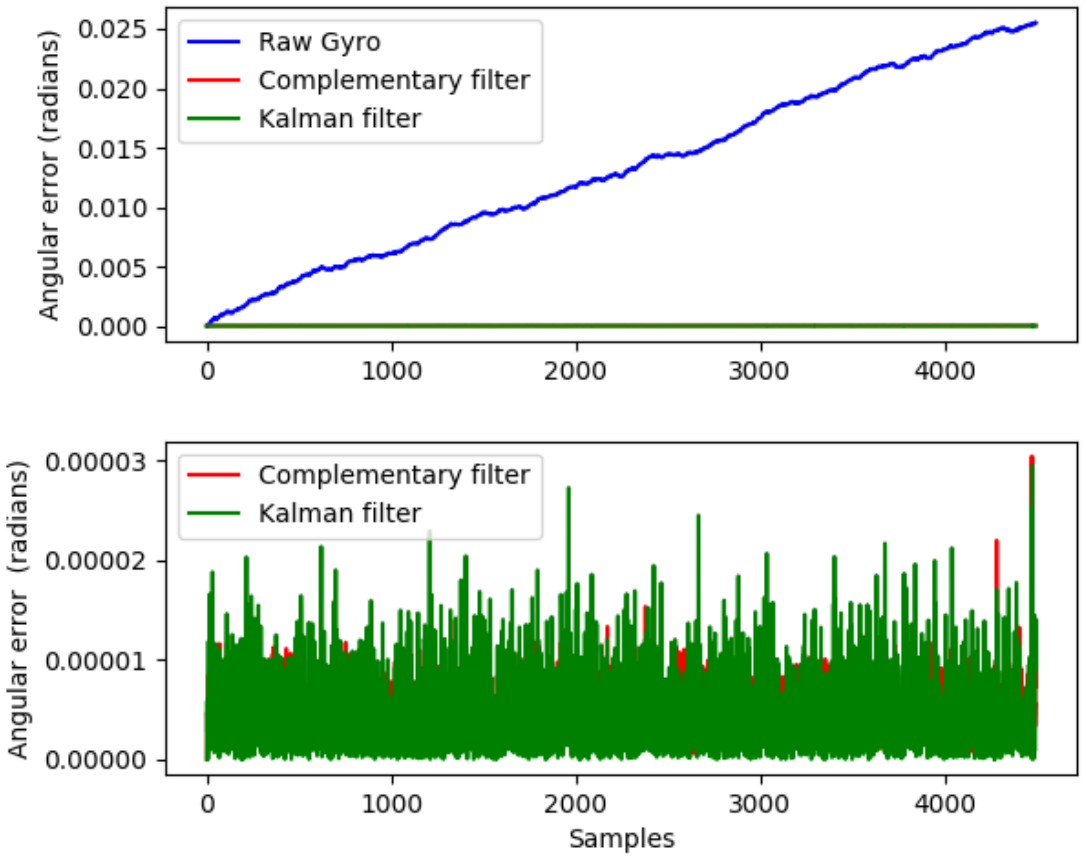


Figure 5.31: Plot of angular error between real and estimated measurement. The bottom plot is the same as the top plot but without the Raw Gyro error.

5.3.3 Effect of video compression

The application of RADES in remote sensing means that there is need for a video compression system. With recent developments in camera technology, a lot of cameras come equipped with an in-built video compression capability. However, there is a need to

investigate the optimum video compression configuration for remote sensing for varying communication channels including wired Ethernet links, point-to-point radio links and satellite links. Other than channel bandwidth, key factors of interests include:

1. Good subjective quality
2. Compatibility with RADES in terms:
    a. Accurate detection of the horizon and hence distance estimation.
    b. Capability of achieving good subjective quality after dehazing.

Some tests have been carried out using uncompressed 720p HD video data gathered during the initial sea trials. The test involved compressing the video to varying number of bits per frame by simultaneously varying the bit rate (bits per second, bps) from 256kbps to 50000kbps and frame rate (frames per second, fps) from 3fps to 50fps. The subjective quality at each bit per frame is recorded as well as the performance of the HoT algorithm with the compressed video.

The outcome of the experiment is that HoT algorithm can work reasonably well with video compressed to as low as 250kbps at 12fps. However, this is dependent on weather. The system performance at lower bits per frame degrades as weather becomes poorer. Although, dehazing improves the performance of horizon detection, it results in poor subjective quality. Figure 5.32 shows the result of dehazing a video compressed to 500kbps@50fps. The blocky artefacts in the sky region are due to compression.

Figure 5.32: Left: Result of dehazing at 500kbps at 50fps; Right: Result of dehazing at 5000kbps at 12fps

As the number of bits per frame increases, the HoT algorithm is better able to cope with poor weather and the subjective quality of dehazing improves. It was also found that, at compression configuration of 5000kbps@12fps, any further increases in number of bits/frame had only a slight effect on performance. Hence, in conclusion:

- For small channel widths like in a satellite link, 500kbps@3fps is a reasonable configuration for RADES to cope with while
- For higher bandwidth networks, 5000kbps@12fps is a reasonable minimum.

To achieve higher frame rate in a small capacity link (especially with dehazing), all processing is done on board the vessel before transmitting over satellite.

5.4 Summary

In this Chapter, a computer vision algorithm for detecting and tracking the horizon was presented. One of the main contributions is the novel combination of several robust algorithms to produce an intricate but effective system that is capable of coping with widely varying environmental conditions at sea. The operation of HoT algorithm is completely automatic and it is tuned for real time operation. These abilities make the system suitable for practical use. The algorithm uses a pre-processing step based on the dark channel prior, which helps with dealing with weather effects and provides dehazing capability in the event of severe weather.

Edge information is obtained from multiresolution analysis based on the premise that edge information tends to stay constant in adjacent scales, while noise does not. A new automated thresholding technique based on hysteresis is proposed which builds on the strengths of two well-known auto-thresholding algorithms. Finally, the horizon position is obtained from singularity analysis in the wavelet domain. The wavelet method proves several times faster than the traditional method based on Hough transform. The algorithm has been tested with several hours of video captured in varying environments and results from sea trials were presented in section 4.3. The algorithm can achieve a processing speed of up to 16fps but this is reduced in severe weather when dehazing is turned on.

Another contribution is a new method for resolving the orientation ambiguity problem associated with the wavelet horizon detection method. This solution provides a metric defining the confidence level of the detection. This proves vital for the operation of the tracking system. An extended Kalman filter operating in the image space was developed to track and predict the location of the horizon in the next image. This simplifies the computer vision process and facilitates real-time operation.

The HoT algorithm is further improved using the cooperation of the camera with a relatively cheap orientation sensor. A loosely coupled data fusion approach was adopted here i.e. the IMU system is run independent of the vision system. A robust quaternion extended Kalman filter for fusing attitude measurement from the camera and IMU has been developed and tested using simulated and real data. This is highly beneficial to further speed up the HoT system and make it robust to temporary occlusion of the horizon caused by vessel motion.

The HoT system facilitates the distance estimation capability of the RADES system described in Chapter 4 and allows identification of the pixel coordinates of the edge of the mitigation zone. The graphic engine stabilises the image from the camera monitoring system using orientation information recovered from the horizon and overlays graphics demarking the edge of the mitigation zone before displaying the result to a screen. The HoT system combined with the RADES algorithm ensures accurate distance estimates to features on the sea surface. However, the technique for detecting features that indicated the presence of a marine mammal on the sea surface is the subject of the next chapter.

Chapter 6 : Towards Automated Recognition of Cetaceans at Sea (ARCS)

Current methods of visually detecting cetaceans on the sea surface rely heavily on human observers using naked eye and binocular to scan the sea surface for features such as blow. These techniques are partially unreliable due to:

1. Limited field of view coverable by the observer at any given time
2. Poor visibility due e.g. to darkness, fog etc.
3. Fatigue due to the repetitive operation of scanning the sea surface.

However, the main advantage of using an observer is their superior ability to intuitively discern whether a feature belongs to a cetacean or not. In this chapter, an algorithm for Automated Recognition of Cetaceans at Sea (ARCS) is presented. The system aims to mimic human observers by detecting features peculiar to cetaceans.

A review of current methods of automated cetacean detection methods based on computer vision was presented in chapter two. Other than one method that relied on multispectral cameras, the other methods used an infrared camera as their primary sensor. It is fairly obvious that the multispectral system can quickly become expensive since it consists of 5 cameras [13] mounted on a stabilisation gimbal. In addition, the automated cetacean detection system in this method seemed to be optimised for aerial images.

The other methods based on infrared camera only detect whale blows since it is a reliable feature easily discernible in the infra-red spectrum. All other features are quite difficult to discriminate in this spectrum because the thermal camera cannot penetrate the sea surface. There is growing interest in this area both for scientific purposes and mitigation purposes.

The algorithm presented here is designed to work in any region including relatively warmer regions where the effect of sea clutter is more prominent. The choice of front-end or sensing system for the detection system is, no doubt, an integral part of the system and has major consequences on its effectiveness. Here, a dual spectral system consisting of a thermal and standard visible camera, as described in section 3.1, is used with a focus on the thermal camera. Relatively cheap off the shelf uncooled long wave infrared (LWIR) thermal camera is the sensor of choice compared to cooled Medium Wave IR (MWIR) used in [15], [16]. The advantage of this type of camera is that it is cheap and does not require servicing whereas cryogenically cooled camera must be serviced every 1-2 years. These cameras are designed to operate in the 7 – 14 microns in wavelength.

The result presented proves that the LWIR camera can produce images of cetacean features that are thermally discriminable from the background with sufficient contrast for automatic detection using a machine learning algorithm. Experimental trials were conducted in

relatively warmer water compared to [15]; approx. 15 degrees. Further experiments in waters of up to 23 degrees in the Azores supports the claim that uncooled LWIR produces sufficient contrast in images albeit with a slight increase in noise as trade off.

The sea environment is very dynamic with constantly changing waves and atmospheric conditions; this provides very complex challenges for computer vision algorithms. In our case, breaking waves are a major source of false positives. Although, a lot of these features are filtered by the feature extraction algorithm, the ratio of the negative set due to waves compared to the positive set (whale blow) is significantly high. Learning from such highly imbalanced data set is one of the main task that is resolved here. Other sources of false positives are birds and the presence of sea weed in the data capture location; these further exacerbates the challenge.

The algorithm proposed by Santhaseelan et al [16] is the most similar to the one developed here. In their work, an elaborate thresholding scheme based on a grid system was proposed. However, the sea environment is constantly changing with different atmospheric conditions caused by fog, mist of constantly varying levels; these can significantly reduce the applicability of this method. Even more so when a less sensitive uncooled LWIR camera is used compared to a MWIR. A feature extraction algorithm that can adapt to the changing scene was adopted here. The parameters of this algorithm can remain constant regardless of environmental conditions, whereas fixed threshold must be tuned as images change with the weather. The extracted feature is then fed to a trained support vector machine classifier. An approach similar to ours has also been applied to PAM [113], they used an active contour segmentation process that adapts to image content and then fed the result to a support vector machine.

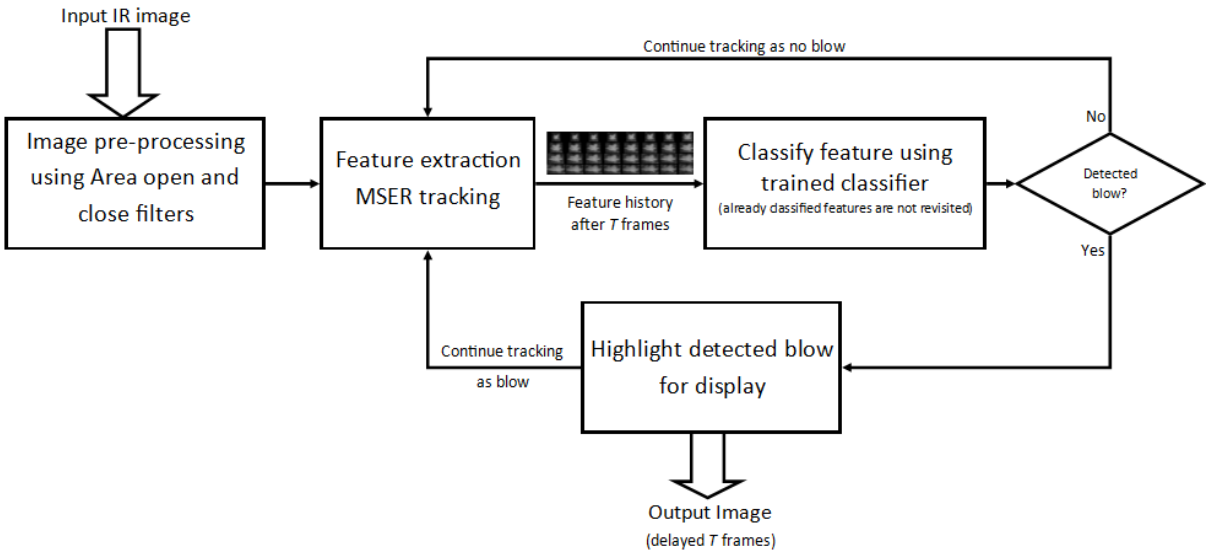


Figure 6.1: Block diagram of the ARCS algorithm

A wide-range of dataset was collected using the adopted front-end system over a couple of days with different sea state and wind conditions. The approach adopted here was to divide the data set into primary data set for development, optimisation and performance analysis

and a secondary data set for testing. The data set was captured in Capetown (mainly southern right whales), from land looking out to sea ensuring that a controlled experiment may be conducted. This area was chosen because of the continuous presence of Southern-Right whales in the bay at that time of the year. It also afforded the opportunity to capture data in, arguably, a more challenging environment compared to what may be obtainable on offshore fixed platforms and vessel because: 1) wind in the bay area was quite strong; 2) rocks and kelp in the area resulted in breaking waves and additional noise sources.

Figure 6.1 shows the block diagram of the algorithm; details of all the steps that make up the new algorithm are given in this chapter. In section 6.1 the feature extraction methodology employed by the algorithm is introduced. The classification analysis used to detect and track cetaceans is presented in section 6.2 and some results are shown in 6.3. It concludes in section 6.4 with a detailed analysis of the system's performance and a summary in section 6.4.1.

6.1 Feature extraction

Feature extraction techniques in computer vision can enable the automatic detection of cetaceans by facilitating the extraction of signature features such as whale blows, breaches and tail flukes in video signals. This is a relatively new area of research and only a handful of works have been done. Here, the aim is to investigate a feature extraction technique or set of techniques that can enable reliable detection or computer-assisted detection of cetaceans. Several commonly used computer vision techniques have been reviewed in section 2.3. The deduction from the review is that, the most promising techniques are those based on deformable feature analysis due to the non-rigid nature of the cetacean blows. Finding the optimal descriptor or set of descriptors that are partially or completely invariant to shape to facilitate a robust real-time detection system is described next.

6.1.1 Pre-processing

The sea environment is a particularly challenging one because the scene is constantly changing with wave and swell appearing and disappearing. The first step in feature extraction is to smooth the image to remove as much noise as possible.

The filtering strategy employed here is based on the premise that the whale blow is always brighter in intensity than its surrounding. To this end, area closing and opening filters were used to smoothen the image in that order. An area close filter refers to a class of morphological filters that operates by removing minima connected regions smaller than a specified area size. The area morphology filters have the benefit that they do not have a fixed structuring element, this enables them to adapt to structures in the image scene. This is in line with the wider strategy employed throughout this section. The reader is referred to [40] for an introduction to area morphology and application in noise removal.

6.1.2 Maximal stable extrema region tracking.

The next step involves segmenting the image to retrieve regions of interests or distinguished regions *(DR)*. The typical approach is to threshold the image into binary class of DRs versus non-DRs. However, finding the optimum threshold value is not a trivial task since it is affected by a lot of variables including sea state, atmospheric condition and depth of the feature to be extracted etc. This led, for example, to the adoption of an elaborate thresholding scheme in [16] that uses varying threshold value at different rows in the image.

Here, a completely different approach that does not require explicit thresholding is adopted. Following the observation made earlier that a cetacean's blow is usually brighter than the surrounding for it to be discernible. A method well suited to extracting connected extremal region available is literature is called Maximal Stable Extrema Region (MSER) [114]. It can be considered as another class of morphological filter that does not use a fixed structuring element, thus allowing it to adapt to features of the image.

MSER operates by first building a component tree of connected regions; each arm of the component tree is then individually traversed to search for stable nodes. The component tree in this context is a representation of a grey scale image that is built by successively thresholding the image with all possible values from 255 to 0. After each threshold, the connected regions in the resulting image forms the nodes of the tree at that level. As the threshold value decreases towards zero, node become bigger in size until there is one remaining e.g. at level 0. If a node change in size after a level change, it becomes the parent of the node at the previous level.

Starting from the top, a node/component at a given level *(L)* is determined as stable by comparing it to its parent at levels (L+Δ) above. The idea is that stable regions tend to remain unchanged over a given relatively large intensity change (Δ). The variation of a component at any level is given as:

$$\psi = \frac{R(+\Delta) - R}{R} \qquad 6.1$$

Where *R* is the area of the region at level *(L)* and *R(+Δ)* is the area *Δ* levels above. A region is considered stable if it has the minimum variation when compared to its local neighbourhood i.e. compared to components above and below. This enables pruning of unstable regions without the need of specifying a threshold value. Thus, the MSER algorithm can be viewed as some sort of an automated thresholding algorithm that requires little or no supervision from the user. The steps described so far are used to detect bright regions (MSER+). Dark DRs can be computed by inverting the image and repeating the above process.

In the work presented here, only MSER+ needs to be computed since blows are typically bright. In addition, a stable component may still be rejected based on some more specified criteria including minimum area, maximum area etc. This makes the algorithm more robust

to noise. One of the major benefits of MSER is that features that describe any given DR can be computed incrementally during the building of the component tree. As a result, no additional computation time is required to obtain properties, for example; area, bounding rectangle, centre of mass, mean grey value, moment information, eccentricity etc.

The original implementation of MSER can be quite slow, and almost impractical for real-time application. A more efficient implementation has since been developed [115] that enables MSERs to be computed in true worst case linear time. The algorithm can be further sped up by incorporating the concept of tracking introduced in [116]. In this thesis, the algorithms are built upon to develop a robust feature extraction algorithm.

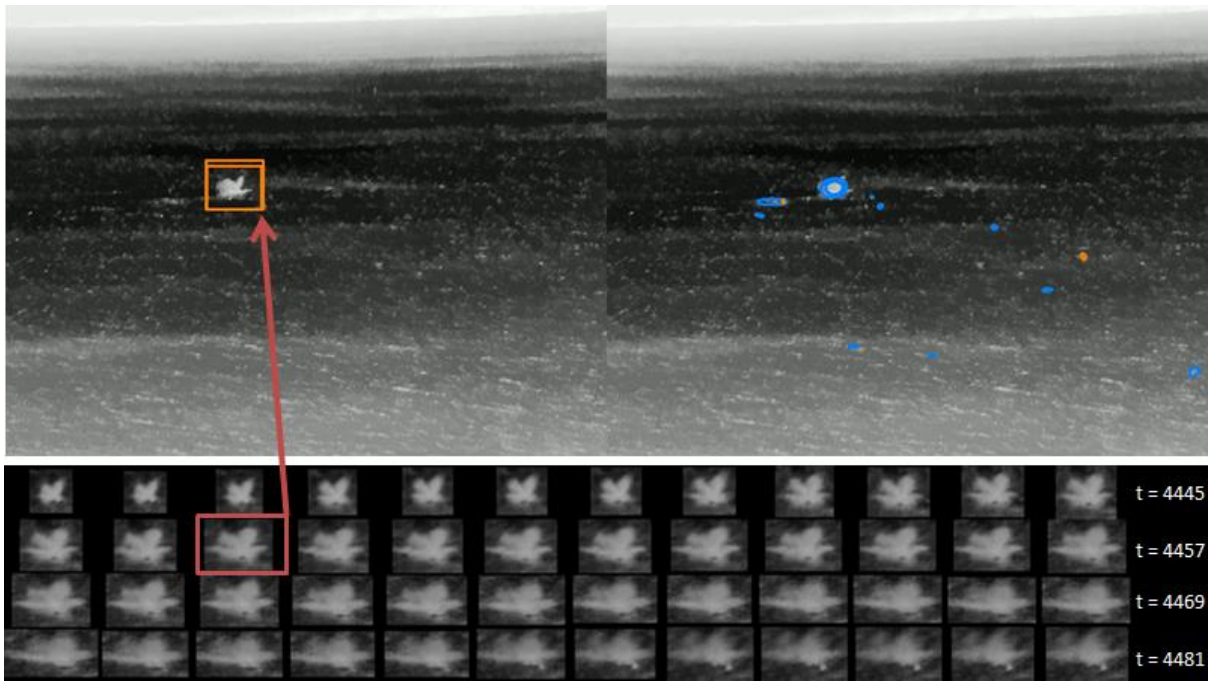


Figure 6.2: Result of MSER tracking of whale blow over time. In this example sequence, the whale blow highlighted on the top-left was first detected in frame 4448 (indicated by the red box in the bottom image); it is then tracked back in the previous 16 frames and then forwards in time until it is lost. Other features tracked in this sequence are highlighted in blue in the top-right.

MSER tracking provides a mechanism for matching detected MSER region in time i.e. for a give DR detected at any given time *t*, a corresponding region is tracked at *t + 1* or vice versa. In addition to speeding up MSER detection, the temporal information has the added benefit of increasing stability of DRs detected when compared to single frame MSERs. For example, a DR can be tracked forwards or backwards in time (e.g. see Figure 6.2); ensuring that almost all regions within the specified parameters are detected while allowing a complete profile to be built over time.

The concept of MSER detection then reduces into a correspondence problem, i.e. matching regions across frame. Rather than searching the entire image for DRs found in the adjacent frame, the tracking mechanism restricts the search to a predefined region of interest (ROI) around the centre of mass of the previous DR. The component tree of the ROI is built and

examined to find the node that best matches the tracking DR. The matching is done by computing a distance score between features of the tracking DR and nodes in the tree. The matching features used here are the same as those suggested in [116] i.e. area, width and height of bounding rectangle, mean and minimum grey value, stability and centre of mass. Other matching / scoring techniques exists in literature e.g. using shape [117], using a tree of shapes,[118] and invariant moments [119]. Shape method is unsuitable here since whale blows do not have a fixed structure, while the moment one is computationally more expensive.

To tune this matching algorithm to the specific application considered here, the features are weighted when computing the distance score using:

$$D_L^{t+1} = \sum_{i=0}^{f} w_i \cdot \left(f_i(R^t) - f_i(R_L^{t+1})\right)^2 \qquad 6.2$$

The node with the best match is the one with the minimum score. The weights have been chosen empirically by analysing a subset of the training data collected. From examining the profile of the tracked blow features, the average change in a feature between frames is computed and the optimal weighing vector is inferred from this. This proved sufficient for the experiment conducted here.

To further improve robustness, if the minimum distance score is above a specified threshold, the region being tracked is considered lost and thus, not tracked further. The size constraint for DR established earlier is also applicable here. However, unlike traditional MSER, the tracked DR does not have to be a stable region. A full MSER of the image is done on the first frame and every *T* frames after. New DRs found at time *nT (for n > 0)* are tracked back in time and forwards.

The technique is very robust when there is little or no camera motion, which is the case in the experiment conducted here. However, this can be easily extended to moving platform by incorporating a motion model to determine the location of regions being tracked. This is discussed further in section 6.4.1.

6.2 Feature classification

Once a DR has been detected and its features are accumulated over time as it is being tracked, the next step is to determine if it is a blow or not. This is the most critical step, and it relies heavily on the quality of the regions detected in the feature extraction step. In this section, a robust model based on SVM is investigated with the aim of real-time classification of DRs. Following review done earlier in this thesis, SVM is preferred due to being mathematically intuitive and simple to implement since multiple features can be concatenated to build a suitable model.

As mentioned, in a classification problem, SVM seeks to find a mathematical model that defines the optimum hyperplane separating two classes. The samples in training set that are closest to the hyperplane are called the support vectors. The optimum hyperplane is one that maximizes the margin *(M)* between the support vectors of both classes:

$$y_i\left(b+\sum_{j=1}^{p} w_j x_{ji}\right) \geq M, \quad i=1,\ldots.n \qquad 6.3$$

Given a set of training samples *(x_i)* and their corresponding classes*, (y_i)* where *$y_i \epsilon$ {-1, +1}*, SVM involves finding the coefficients *w* and the bias *b* such that $b+\sum_{j=1}^{p} w_j x_j = 0$ for samples that lie on the separating hyperplane and the distance of the hyperplane from the origin is given $|b|/\|w\|$ where $\|w\|$ is the Euclidean norm of vector ***w*** [120]. Similarly, we can write an expression for the hyperplane formed by the positive support vector as $b+\sum_{j=1}^{p} w_j x_j = 1$ with a distance $1-|b|/\|w\|$ from the origin. From this relationship, it is obvious that the margin *M* is $1/\|w\|$. SVM then becomes an optimisation problem that seeks to minimize $\|w\|$. Minimizing $\|w\|$ is the same as minimizing $\|w\|^2/2$ which makes it possible to perform Quadratic Programming (QP) optimization [121].

Such a hyperplane, however, is only applicable for linearly separable samples. To cope with non-linear problems, the kernel trick [121] is used. This involves using a function *φ* that maps the input sample to a high dimensional space. Equation 6.3 becomes:

$$y_i\left(b+\sum_{j=1}^{p} w_j\,\varphi(x_{ji})\right)-1 \geq 0, \quad i=1,\ldots.n \qquad 6.4$$

with

$$\sum_{j=1}^{p} {w_j}^2 = 1$$

The support vector machine described so far ensures that all training samples lie on the right side of the hyperplane. In reality, the training sample usually contains noise and this strict criterion can result in overfitting. To cope with this, a parameter *C* is introduced to allow some samples to be on the wrong side of the hyperplane such that:

$$y_i\left(b+\sum_{j=1}^{p} w_j\,\varphi(x_{ji})\right) \geq 1-\xi_i, \quad i=1,\ldots.n \qquad 6.5$$

with

$$\xi_i \geq 0 \; and \sum_{i=1}^{n} \xi_j \leq C$$

6.2.1 Whale blow characteristics.

Figure 6.3 shows the result of a blow event found using the feature tracking algorithm. Its area and height are smoothed with a Gaussian filter before plotting. It shows that for a fully formed whale blow, some of its features including area and height will increase for a period before starting to diminish. The duration of the rise and fall times varies widely from blow to blow due to size of the animal (i.e. adult or calf) and sea conditions (e.g. wind speed, swell height etc.). A partially formed blow tends to have a partially formed Gaussian curve due to aforementioned reason; the example shown in Figure 6.4 was due to the wind.

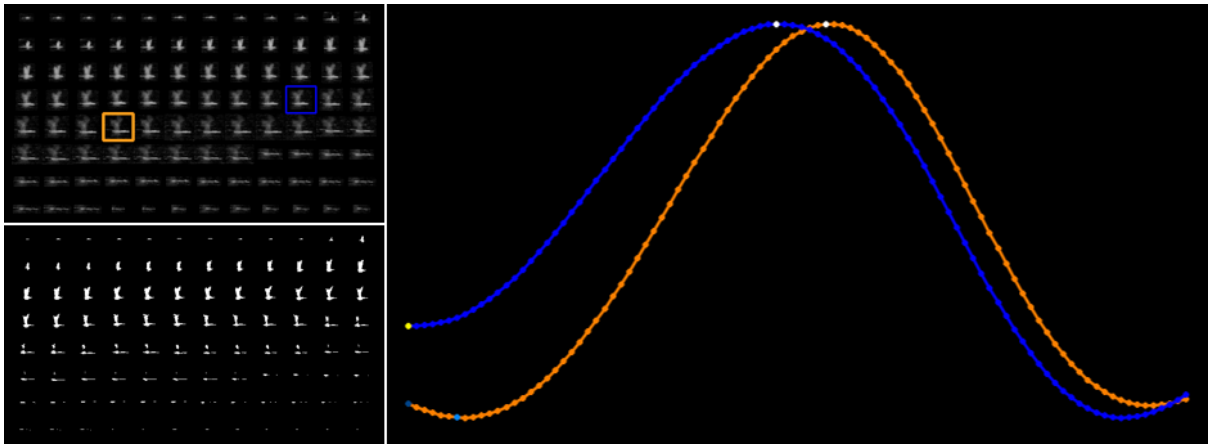


Figure 6.3: The result of the feature tracking algorithm for a fully formed whale blow. Top left: Whale blow tracked with the highlighted frames corresponding to the white on the corresponding plot of same colour; Bottom left; binary image of tracked region to show whale blow more clearly; Right: Plot of DR area (orange) and height (blue) against time.

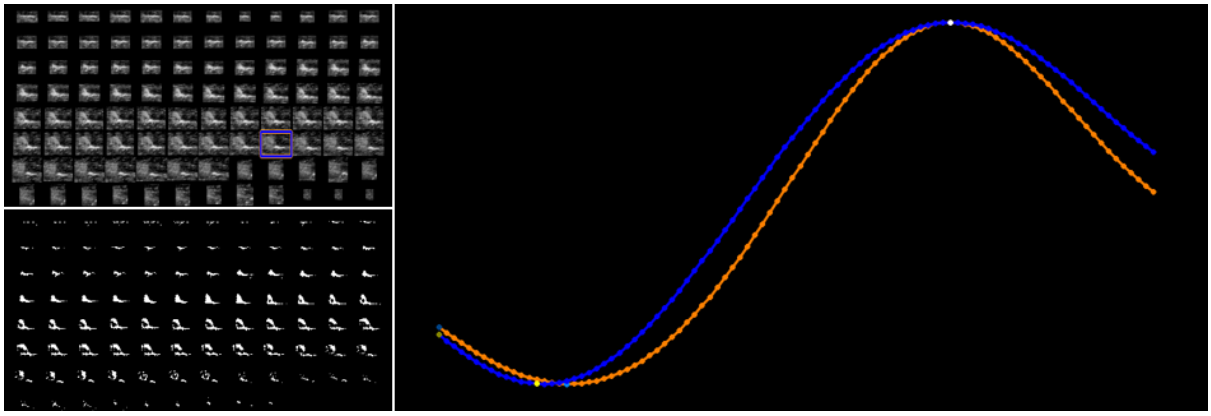


Figure 6.4: The result of the feature tracking algorithm for a partially formed whale blow. Top left: Whale blow tracked with the highlighted frames corresponding to the white dot on the corresponding plot of same colour; Bottom left; binary image of tracked region to show whale blow more clearly; Right: Plot of DR area (orange) and height (blue) against time.

Wind causes the blow structure to disintegrate very quickly after it is formed, hence the fall time is relatively smaller than the rise time. The rate of disintegration also results in drastic changes in the region characteristics thereby reducing the ability of the feature tracker to track it effectively. To cope with these scenarios, we investigate only the rise time of the whale blows. The rise time calculated for blows found in the training data set running at 24fps ranged from 9-34 frames.

Taking the duration of the whale blow to be 2 times the rise time, suggests that the feature is only visible for 0.5 – 3 seconds. This wide range of blow duration further emphasizes the enormity of the challenges faced here. To ensure that the rise time of any blow is fully observed, 48 is the minimum number of consecutive frames (2 seconds) that must be assessed first. For any event with duration d (d ≥ 48), only the first 48 frames are considered while those that last for less than 16 frames are ignored. The features of any events with duration *d* frames (16 ≤ *d* < 48) are resampled (upsampled) to 48 frames using bi-cubic interpolation. This makes it easier to compare events to each other. For each event, *n* features are extracted yielding an $[n \times 48]$ dimensional descriptor; in this thesis *n=35*.

6.2.2 Data Balancing and pre-processing

Using equation 6.5 and a set of training samples, an SVM model of whale blow may be developed. To get the optimum result, a few pre-processing steps are required.

Data Normalisation

To develop the SVM model, *n* multiple feature vectors each having *t* dimensions are concatenated to yield one vector *[1 x tn]*. It was established in section 6.2.1, that the optimal value of *t* is 48. However, some of the *n* concatenated features may have different ranges for example, grey level is [0 - 255] while eccentricity is [0 – 1]. This results in a problem whereby features with a large dynamic range dominates the model even though they might not be the most effective feature. To eliminate this bias, the features must be scaled before concatenation in a process termed standardisation. The aim of this step is to ensure that all features have similar dimension or ranges and also eliminate any numerical problems that might occur due to large numbers [42].

There are several standardisation techniques that exist in literature and the best technique usually depends on the problem. In SVM, the most common method used is Z-score normalization and thus adopted here. It involves standardizing feature x_j using the variance σ_i and mean $\tilde{x}_i$:

$$\hat{x}_i = \frac{x_i - \tilde{x}_i}{\sigma_i} \qquad 6.6$$

Data cleaning and balancing

A naïve implementation of SVM performed very poorly due to highly unbalanced dataset. Imbalanced data is intrinsic to the problem due to the dynamic nature of the sea environment. The complexity of the data is further exacerbated due to within class imbalance caused by 1) the varying nature of the positive class (the blows) and majority of the negative class (e.g. waves), 2) overlapping caused by other events such as tale slap, breaching etc. The flow chart in Figure 6.5 gives a detailed overview of the training process.

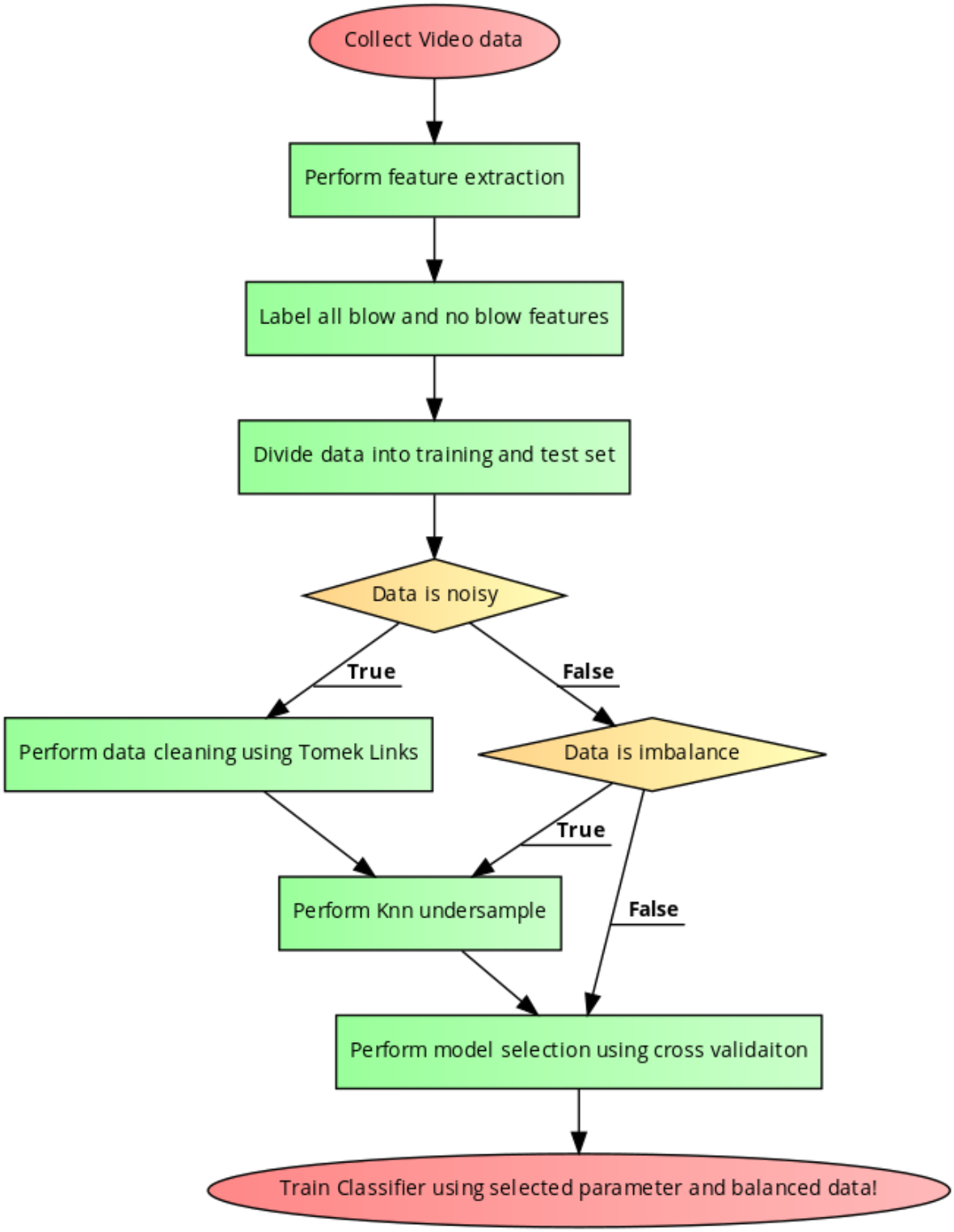


Figure 6.5: Flow chart showing the steps involved in the training of the SVM classifier.

Learning from imbalanced data is still a focus of intense research [122]; a lot of work has been carried out in this area [123]–[125] and the most common methods are based on resampling the dataset. The easiest resampling method is based on random oversampling of the minority class or undersampling of the majority data. Oversampling is not considered here since it will result in a data set that is too large and could take an infinitely long time to

train. Besides, it is well documented in the literature that oversampling often results in overfitting e.g. as stated in [125]. However, experiments showed that random undersampling was inconsistent due to the random nature of the process. Majority of the time, the model performed poorly perhaps due to selection of too many overlapping samples or too many samples distant from the blow samples.

Another technique other than resampling that has been employed in literature is based on data generation. Again, data generation will mean upsampling the minority data set to balance the majority set. Since oversampling is inconvenient, data generation is not explored further, and it becomes clear that an effective undersampling technique needs to be identified and implemented.

This led to the investigation into informed undersampling techniques. This techniques overcomes the information loss caused by the traditional method of random undersampling [123]. This technique involves careful and deliberate selection or removal of samples from the majority class using a predefined criterion. The method of deciding informative samples is critical in this approach.

The first method considered here involves partitioning the majority data into *n*-balanced partitions. Each partition is combined with the minority class in succession to train and build an SVM model. The majority class samples that are used as support vectors in each model are kept as informative and others discarded. This process is then repeated until the required number sample number is reached or when the algorithm converges. Unfortunately, this method resulted in a model that performed poorly due to overfitting. Another disadvantage of this method is that it takes long time to resample since it involves training an ensemble of SVMs. The algorithm may also converge too early. In this case, random undersampling is used to discard excess samples.

An alternative approach is to use K-nearest neighbours to select informative samples as described in [124]. This simply involves computing the similarity between a sample in the majority set and a minority set using a distance measure such as Euclidean distance. Once the distances / similarities are computed for all majority samples, they are ordered and the *k* samples nearest to the minority samples are selected. Again, this resulted in an overfitting model. However, skipping the first *q* nearest samples and selecting the next m nearest sample gave much improved results. Experiments showed that the value of *q* and *m* chosen had a significant impact on the ability of the model to distinguish classes. Based on the results achieved using the KNN and the ensemble method, it becomes clear that the problem is not necessarily an imbalanced problem but a noise problem due to between class overlapping thus causing overfitting. This led to investigations into sample cleaning techniques.

For every sample x_i in any class, the distances to the nearest neighbour in the opposite class $d(xi, x_k)$ is computed and the distance to the nearest neighbour in the same class $d(x_i, x_j)$ is

also computed. If $d(x_i, x_k) < d(x_i, x_j)$ then clearly the majority class sample in the pair is a noisy sample that may cause overfitting if left in the class; x_i and x_k are called Tomek link pairs [123]. The Tomek link is broken by removing the majority sample that makes up the pair, this process is repeated until all links are broken. K-nearest neighbour undersampling is then performed after cleaning.

This method gave the best performance and is the balancing technique adopted here. Note that, equal number of samples are not selected for both classes as may be expected; instead the negative class sample is undersampled to *r* times the positive class. In this thesis *r* is equal to 4 and this further proves that a small amount of imbalance is not a problem for SVM but overlapping and noisy data set is.

6.2.3 SVM modelling

Having prepared the training data, the next step is to train the SVM model but first, a decision must be made about the kernel function to use and the value of the parameter *C* that controls how strictly the model fits the data. One of the most common kernels is the Radial Bias Function (RBF). It is very suitable for non-linear problems and has fewer numerical difficulties when compared to polynomial and sigmoid kernels [42]. The RBF function is:

$$K(x_i, x_j) = e^{-\gamma\|x_i - x_j\|^2} \qquad 6.7$$

Next, the optimal value of kernel parameter γ and the tuning parameter *C* must first be selected. A very simple way of searching the parameter space to try different values until the best result is achieved. This is typically done by splitting the training set into two subsets, the learning set and the validation set. Multiple SVM models are trained using the learning set while continually varying the parameters we are searching for. The model that performs best on the validation set is then selected.

Cross-validation

The concern with this strategy is that, there is limited amount of training data available and so the validation set is usually much smaller than the learning set, as a result the parameter search is biased towards the validation set and may not generalize the problem being solved. A more widely used strategy for parameter selection in literature called cross validation is adopted instead. Cross validation seeks to eliminate any bias during model selection by performing multiple rounds (e.g. k-folds) of learning and validation. In each round, the samples from the training set used for learning and validation is continually varied. The performance of each model is taken as the average for all rounds.

If the validation and learning set are chosen randomly in each round, Random cross-validation is said to be performed. Using this technique, it is possible that some samples are never used for validation. To ensure that all samples are used at least once for validation, k-

Fold cross validation can be used. This involves splitting the training data into k-sets, each of the k-set are then used in turn for validation while the remainder are used for learning. Each technique has its advantage but the preferred here is k-fold cross validation, as recommended in [42].

6.3 Sea trials

A detailed description of the feature extraction technique has been given in section 6.1 and all the steps required to develop a robust classifier has been introduced in 6.2. In this section, preliminary trials conducted to evaluate the algorithm done in Gansbaai South Africa (34.5849117S, 19.3348983E) are presented. The thermal camera was mounted on a tripod overlooking the sea on the balcony of the "whale house" used for the research. Visual observers were employed to monitor the images of the camera day and night and record sightings over a three (3) day period.

This area was chosen due to a high population of Southern Right Whales that inhabit these nearshore waters during the South African winter months (May – October). The temperature of the water was determined to be around 15 degrees, using sea surface temperature data obtainable from NOAA (National Oceanic and Atmospheric Administration). A long wave infrared camera was used and video from the camera was recorded to files with duration of about 3.5 minutes each.

Table 6-1: Table showing file set used for training and testing SVM classifier

	No of files	No of blows
Training and Validation data	10	22
Test data	15	32

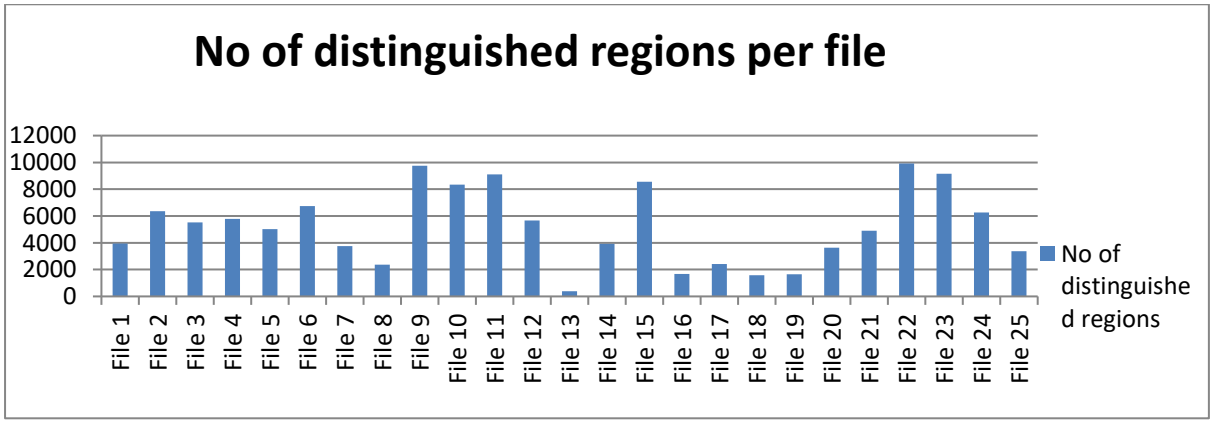


Figure 6.6: The number of distinguished regions detected in each of the files considered. The files are split into training and test set using this number to ensure that the training data is a good representation of the dataset.

The classification algorithm is then developed using the flow chart in Figure 6.5. Feature extraction was performed on all the files and files were ranked based on the total number of features that were tracked in them as shown in Figure 6.6. 40% of the files were chosen for training and validation and the remaining 60% for testing. The files were split based on their

ranking to ensure that the training sample set was a good representation of the entire data set. This resulted in 22 blows events in the training data set and 32 in the test data set as detailed in Table 6-1. It is important to note that the amount of DR detected in a file seems to be directly related to sea state as well as time of day. More DR were detected during the day due to increased emitted clutter caused by reflection from the sun. Increase in breaking waves in high-sea state also resulted in increased emitted clutter.

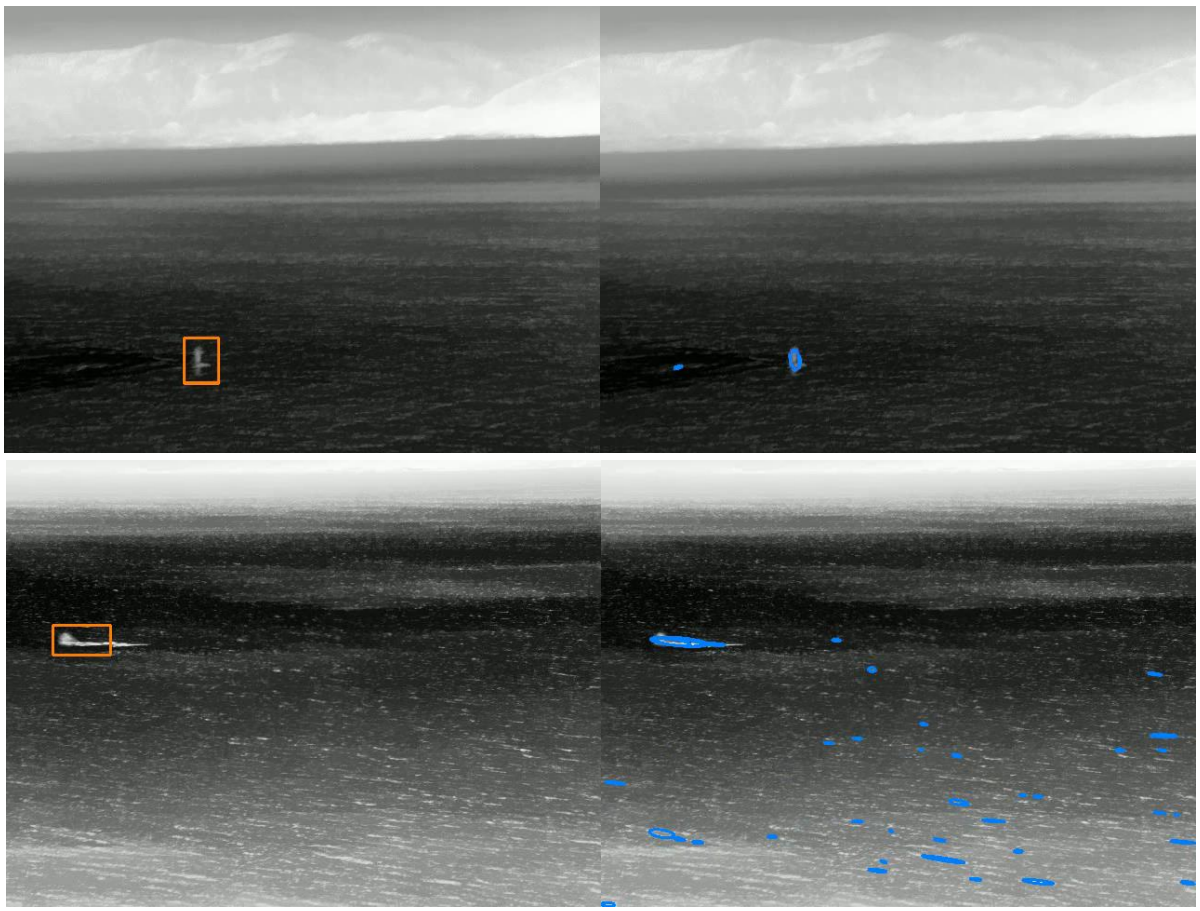


Figure 6.7: Whale blow detected by SVM model. Top Left: Fully formed whale blow detected; Top Right: MSER regions tracked in time; Bottom Left: Partially formed whale blow detected; Bottom Right: MSER regions tracked and correctly filtered out by SVM model. The orange boxes on the left are detected blows while the blue ellipses on the right are MSER regions being tracked in time.

Result shown in Figure 6.7 proves the MSER tracking algorithm is very robust and that the SVM model was able to generalize the problem. The SVM was effective at detecting whales blows dissimilar in appearance to those used to train it. The model was able to predict 87% of the blow events correctly in the test data and detect 95% true negative events.

The major sources of noise can be divided into non-cetacean features and cetacean features as shown in Figure 6.8. Non-cetacean features present in the sea that caused false positive detection include sea birds, kelp (sea weed) and waves. Cetacean features caused by breaching, tale slap, fins and the animal breaking water all caused false positive detection. Although, non-cetacean feature constituted the majority of false positive detection. However, those caused by cetacean features are desirable since this facilitates the detection of animals before they blow or continual detection/tracking after blowing.

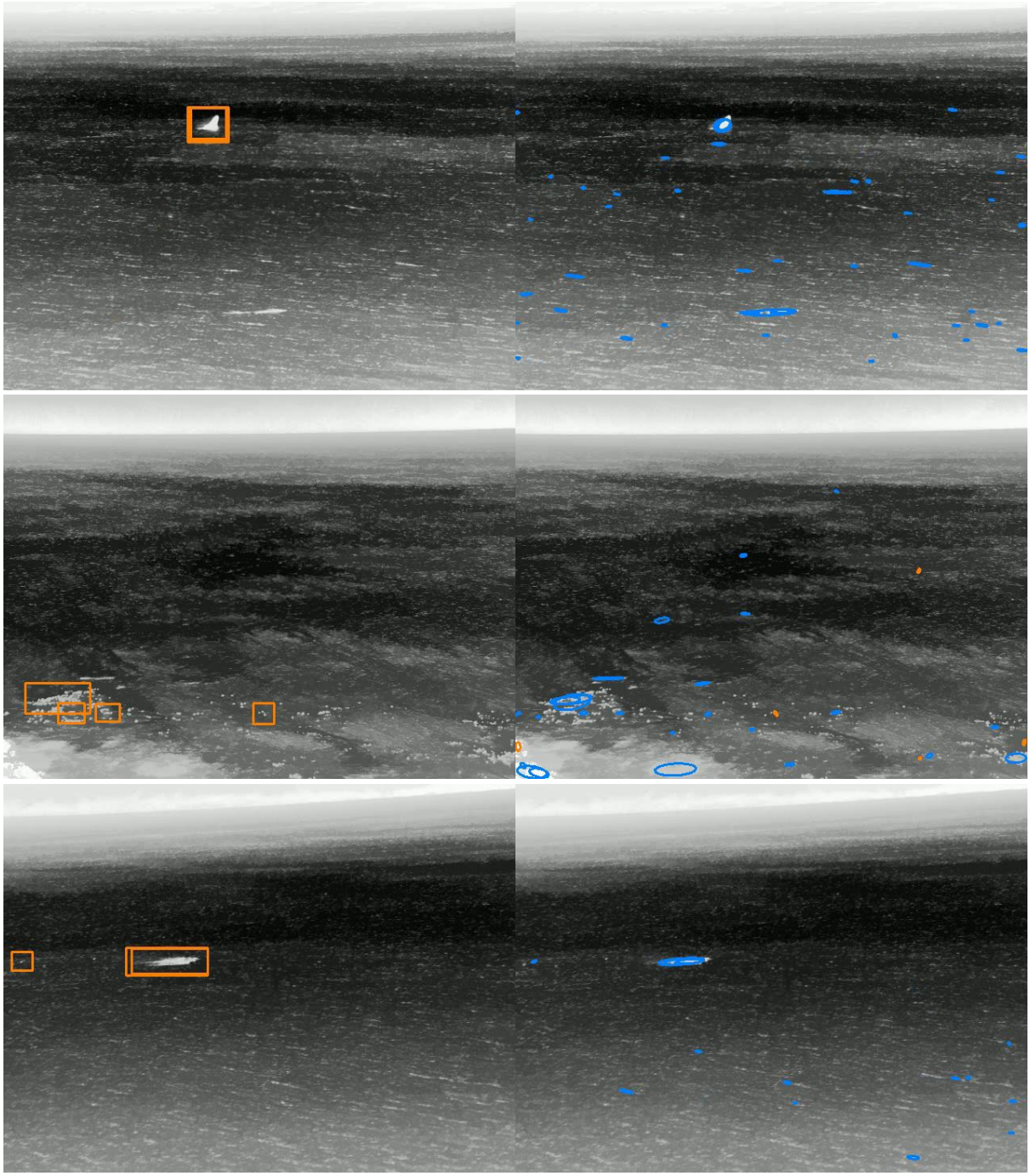


Figure 6.8: False detections by SVM model. All images on the left show regions classified as positive by the SVM model marked by the orange box while images on right show all MSER region tracked and classified by the model. Top: Whale breach; Middle: Sea weed - kelp; Bottom: Bird and whale footprint.

6.4 Performance analysis and Discussions

As show in previous section, the classifier does perform very well while eliminating majority of true positives. The result here has proven that LWIR thermal camera can produce images of cetacean features that are thermally discriminable from the background with sufficient contrast for automatic detection using machine learning algorithm. In this section, analysis of the ARCS software is presented with focus on:

1) Quantitative analysis using a Receiver-Operator-Characteristic curve and qualitative analyses by comparing to traditional method i.e. MMO.
2) Analysis of best feature to use for Tomek link analysis
3) Theoretical analysis of achievable detection distance.

Experimental sea trials and Evaluation of system performance

The main criteria for success is achieving similar detection rate compared to MMOs who are "industrial standard". Videos were analysed by experienced MMOs to collect ground truth data with which the result from ARCS was compared. In addition to this, ROC curve analysis was also conducted to compare the result achievable by the SVM classifier with and without sample cleaning using Tomek links. The result shown in Figure 6.9 clearly shows that the classifier performs better, achieving better true positive rate when sample cleaning was performed.

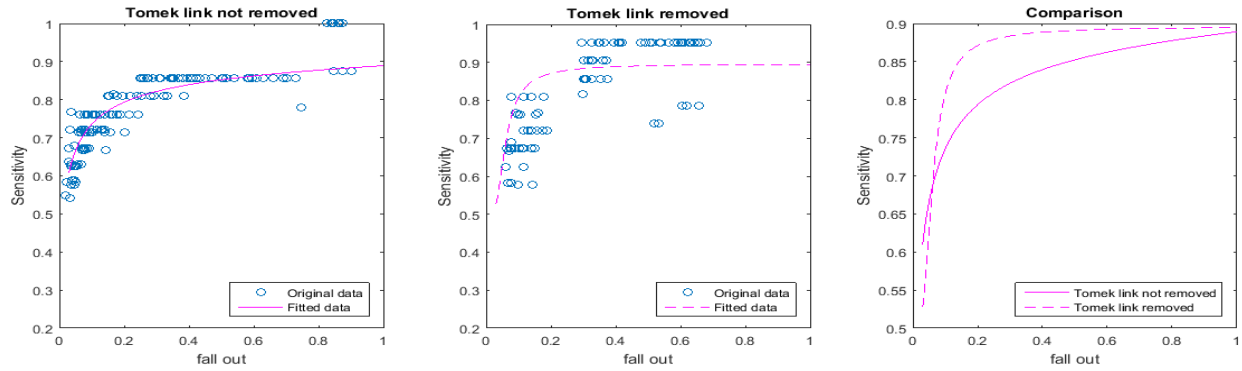


Figure 6.9: ROC curve analysis of SVM classifier for the unbalanced dataset shows that better performance is achieved when data cleaning is performed prior to undersampling.

Tomek link filtering

As already establish, this is a crucial step to prevent overfitting by the SVM model and the preferred method used here is based on finding and breaking Tomek link pairs. In this section, we perform analysis to determine which feature space is used for data cleaning using Tomek links.

A naïve implementation of this process may include finding Tomek links using each individual feature to clean the training set before training. However, analysis shows that there is no correlation between Tomek links found using one feature compared to another, thus, such naïve implementation may result in removal samples highly beneficial to certain feature. Another approach is to rank features based on the number of Tomek links detected in its space.

The feature with the fewest Tomek links in its space may suggest that it has superior discriminating power and thus ranked higher. This analysis was performed on the training data set and result is as shown in Figure 6.10. To test the stability of this ranking mechanism, k-fold cross validation is employed. The idea is to evaluate each feature when

some of the training set is removed. Result shows that the top 8 features were the similar before and after cross-validation. One of which includes the concatenation of all *(t)* features of size *(n)* to form a *[1 x tn]* feature. This is the feature used for all Tomek link analysis and data cleaning steps for the SVM classifier developed in this thesis.

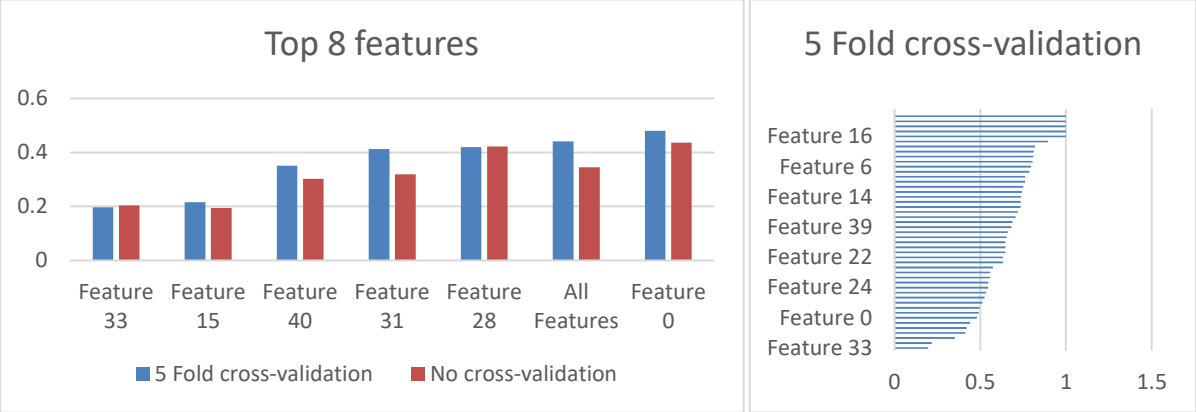


Figure 6.10: Feature ranking based on Tomek link. This result shows that the top 10 ranking features are consistent with and without cross-validation. To this end, "All feature" is selected for Tomek-link data cleaning step.

Analysis of achievable detection distance

As part of the work here, the theoretical detection distance that can be achieved is analysed. This will enable the selection of the right camera parameters e.g. in terms of resolution and FOV. Recall that;

$$FOV = 2\ \tan^{-1}\left(\frac{D_{pixel} \times X_{mm}}{2 \times x_{pixel} \times Z_{mm}}\right) \qquad 6.8$$

Thus,

$$x_{pixel} = {}^{K}/_{Z_{mm}}$$

This means that pixel size of any target in an image is directly proportional to the inverse of its distance from the camera where *K* equal to $(D_{pixel} \times X_{mm} \times 0.5)/\tan(FOV/2)$ is a constant for a given camera and target size. If the minimum blow size that is detectable by the classifier is known, it becomes trivial to determine the theoretical maximum distance at which a blow may be detected. The camera used in this analysis has a FOV of 15° and a pixel size of 800. In this thesis, 10 pixels is chosen empirically as the minimum required to detect an animal, thus a range of more than 4000m can be achieved for a target about 9m in size; grey whales are widely regarded as the biggest of the cetaceans and they have tall slender vertical blow 9-12m in height. Detection range will be reduced for smaller animals but using a narrow field of view camera can help in this instance.

Other factors that affect the range of detection is the height above sea level and its sensitivity. For example, at 0.5m above sea level, the distance between the horizon and the camera is around 2500m; this effectively becomes the maximum detection range. Although detection range of the ARC software is not directly affected by camera height as shown by equation 6.8, it does affect what is visible.

Figure 6.11: Effect of fog. Top Left: thermal image of fog in early hours; Top right: corresponding visual image of the same scene at same time as thermal camera. Bottom Left: thermal image of same scene after 4hrs; Bottom right: corresponding visual image.

The thermal camera is like any other visual sensor and its operation is governed by the beer-lambert equation on atmospheric scattering which states that the intensity at the observation point is a function of the exponential of the distance from the source and some absorption coefficient. Environmental conditions such as fog, sea mist and spray etc. are all factors that affect the value of the absorption coefficient and thus the magnitude of intensity received by the cameras detector. Some of the data collected have shown that quality of thermal images can be severely comprised by fog as shown in Figure 6.11. It is well known that the performance of thermal camera is no better than the visual camera in this situation. It is obvious that the range of ARCS will be limited in these cases. However, no systematic experiment has been conducted to evaluate the sensitivity of the uncooled thermal camera employed here compare to cooled camera to determine its effect on range of detection.

6.4.1 Feature tracking for a moving camera using RADES

The RADES software introduce in Chapter 4 and HoT algorithm in Chapter 5 can be used to develop a motion model for the feature tracking algorithm. Equations for determining the orientation of a camera from horizon position have been derived in equations 4.8 to 4.13. However, it is impossible to deduce the rotation of a camera about its y-axis from the

horizon alone and as a result, measurement from external sensors such as IMU must be relied upon. Once the orientation of the camera is known, then the rotation matrix R_t at time *t* can be built as in equation 4.8.

The coordinate system introduced in section 3.3.1 is adopted here. Let R_{CG_t} be the rotation at time *t* that takes a point *P* in the world coordinate to the camera coordinate, we can re-write the projection equation in 4.8:

$$p_t = K\left(R_{CG_t}(P + T_t\right) \tag{6.9}$$

At time *t+1*

$$R^T_{CG_{t+1}}K^{-1}p_{t+1} - T_{t+1} = R^T_{CG_t}K^{-1}p_t - T_t \tag{6.10}$$

If the camera motion is dominated by pure rigid rotation, the translation $\boldsymbol{T}_{t+1} = \boldsymbol{T}_t$. Hence:

$$p_{t+1} = K \cdot R_{CG_{t+1}} \cdot R^T_{CG_t} \cdot K^{-1} \cdot p_t \tag{6.11}$$

Since the IMU to camera rotation is known (see section 3.3.2) and the IMU rotation can be directly measured from the sensors, the camera rotation can be easily estimated and equation 6.11 becomes:

$$H_{t+1} = K \cdot R_{CB} \cdot R_{BG_{t+1}} \cdot R^T_{BG_t} \cdot R^T_{CB} \cdot K^{-1} \tag{6.12}$$

$$p_{t+1} = H_{t+1}p_t \tag{6.13}$$

H defined as the homography matrix that maps a point in the image at time *t* to its new location at *t+1* and $\tilde{R}_{CG} = R_{CB}R_{BG}R^T_{CB}$. For a DR detected at time *t,* with its centre of mass located at p_t, the ROI used for correspondence matching at time *t+1* must be located at point p_{t+1} found using equation 6.3.

6.5 Summary

A new automated cetacean detection algorithm has been presented. One of the main contributions comes from the unsupervised application of MSER tracking for feature extraction. Compared to previous methods based on fixed threshold, this technique does not require constant tuning with changes in environmental conditions. The parameters of the MSER detection algorithm determines the minimum and maximum blow size considered by the ARCS algorithms and the contrast of the feature against its background regardless of weather effects. In addition, the tracking capability of the MSER technique allows features to be tracked forwards and backwards in time, ensuring that all features of interest are captured.

The output of the feature tracking algorithm is a set of spatial and temporal properties defining the extracted region which is fed into an SVM classifier trained from a highly imbalanced data set. The method for dealing with the imbalance includes cleaning the dataset using Tomek link analysis followed by a k-nearest neighbour undersampling

technique. This step proves vital in the modelling of the classifier as shown by the ROC curve analysis. The analysis clearly demonstrates that the classifier performs better, achieving greater true positive rate when sample cleaning was performed.

The data set used was divided into two sets; one for training and validation and the second for testing. The result of the test shows that the algorithm is achieved 87% accurate detection of true positive while filtering out up to 95% of false positives. It is also capable of detecting other cetacean features including tale-slap and breaching. For use on a moving vessel, a simple motion model has also been proposed for tracking features with orientation information recovered from the Horizon Tracking (HoT) algorithm presented in Chapter 5.

The choice of front-end system is integral in the development of the algorithm and here, the relatively cheap long wave infrared (LWIR) is the sensor of choice compared to cooled Medium Wave IR (MVIR) used in [15], [16]. The result here has proven that LWIR thermal camera can produce images of cetacean features that are thermally discriminable from the background with sufficient contrast for automatic detection using machine learning algorithm. In addition, data was collected in relatively warmer water compared to [15]. Further experiments in waters of up to 23 degrees in the Azores has supported the claim that LWIR produces sufficient contrast in images albeit with a slight increase in noise as trade off.

Finally, long term testing of the ARCS algorithm on a vessel e.g. during a seismic operation has not been possible in the work presented, due mainly to lack of time. Future work that can be done to improve the work done so far on the RHVM system and a summary of the main contributions of this thesis are given in the next chapter.

Chapter 7 : Conclusion and Future Work

In this thesis, a detailed description of the design and development of a new visual monitoring system for marine application is described. While the focus of the work here is on the particular case of marine mammal monitoring, the system can be easily extended to other applications such as surveillance, estimating distance to nearby vessel or boat etc.

The complete RHVM consists of several hardware and software components described in previous chapters; including: a) CMS introduced in Chapter 3 and b) RADES and ARCS described in Chapters 4 – 6. The new system offers several improvements over traditional methods of visual monitoring including:

1. It facilitates 24hr (day and night) visual monitoring enabling survey operation to continue in low-light conditions
2. Enables accurate estimation of range to targets reducing costs associated with survey downtime
3. Computer assisted detection of targets reducing risk of exposing animals to high intensity noise
4. Objective recording of video data showing targets and estimated distances
5. Reduced health and safety risks since monitoring can be done from remote location
6. The solution is flexible and relatively cheap since it does not require expensive stabilisation platform and uses commercially available off the shelf sensors

This chapter summarizes the main contributions of the thesis in Section 7.1 and outlines future work to be done to further improve the proposed solution in Section 7.2.

7.1 RHVM System

To make the system affordable and flexible, the hardware system was designed using various commercially available components. This posed a challenge to sensors data fusion due to sensor misalignments, both in space and in time. A simple but efficient calibration system was developed specifically to resolve this issue, as presented in Chapter 3. The main advantage of the calibration system is that it does not require any elaborate rig or platform, instead it utilizes a simple calibration pattern that is easy to manufacture. A special planar pattern that is suitable for both visual and thermal cameras was adopted. Spatial calibration is performed by observing the vertical direction in each sensor and estimating the rotation that brings them into alignment. For an IMU, vertical direction is obtained directly from accelerometer readings while for a camera, it is related to the vanishing points of vertical structures in observed in the calibration pattern. Temporal calibration is a result of correlating the angular measurements from the sensors when put through the same

rotating motion. Calibration only needs to be performed once for the fixed group of sensors that make up a CMS. Once the spatial and temporal parameters have been recovered, they remain relatively constant.

In addition to the methods that rely on a calibration pattern, a new online calibration algorithm was presented for use at sea. This solution is applicable for calibrating a camera rigidly attached to an IMU of known spatio-temporal misalignment or conversely estimating said spatial-temporal misalignment given a calibrated camera. This is particularly useful to allow camera lens parameters to change during deployment at sea e.g. zoom in and out to investigate features of interests. The technique is based on the application of the horizon line detected in an image to estimate the vertical direction eliminating the need for a calibration pattern. Compared to a previous method that requires accurate positioning of the calibrating pattern during spatial calibration, the use of a naturally occurring feature in the online method eliminates such requirements. It was also shown that camera intrinsic parameters cannot be recovered from observing the horizon alone and additional information e.g. from an IMU is required; which proved robust. Accurate camera parameter estimation using the above methods is vital for the distance estimation algorithms.

The distance estimation algorithm called RADES is based on simple geometry using the camera pin-hole model. All mathematical equations have been derived from first principles with no simplifying assumptions. Compared to previous work that can only be used offline (since it required manual identification of the horizon), RADES is completely automated and operates in real-time. In addition, we have conducted an extensive theoretical and practical analysis of the method. Analysis shows that resolution in pixels/meter depends on parameters such as camera field of view and the size (in meters) of the area to be monitored. A mathematical expression relating these parameters to the resolution has been formulated in Chapter 4. Hence, the system must be configured with the right parameters to achieve the desired resolution for a given application. The RADES algorithm enables graphics to be drawn on the image in real-time to demark the mitigation zone. This has the advantage that objective and recordable assessment can be made by operators to determine if a target is in the mitigation zone or not. The algorithm requires the orientation of the vessel to be known, in this thesis a sensor fusion approach based on horizon tracking was developed.

The new horizon tracking system presented in Chapter 5 known as HoT consists of two aspects; 1) computer vision algorithms for tracking the horizon in video signal and 2) a robust filter that combines measurements from visual system with Inertial sensor measurements when available. A selection of widely used computer vision algorithms for horizon detection have been reviewed. However, the specific problem of a real-time automated systems at sea capable of dealing with complex challenges caused by continually changing atmospheric condition and noise due to waves has not been effectively dealt with in literature. In this thesis, a pre-processing step based on the dark channel prior was proposed to deal

with these challenges; this proves to be more robust than commonly used denoising techniques such as Gaussian blur. This step also facilitates haze removal with the benefit of enhancing visual range and reducing the effect of mist often present at sea.

A well-studied and established multiresolution technique based on wavelet transform was used for edge detection and a simple thresholding scheme based on hysteresis was proposed to deal with varying illumination conditions. The scheme combines the advantage of the multimodal Otsu thresholding scheme with the unimodal method proposed by Rosin. While Otsu's method is ideal in good conditions, Rosin's is ideal in dull weather, thus combining these two methods provides an optimum solution. The last step is horizon localization, which also relies on wavelet transform. This technique is based on analysing the row distribution of edge pixels. Compared to well-known techniques such as the Hough transform, it has the advantage of being fast which is ideal for real time application. The horizon detection system also includes a Kalman filter for tracking the horizon in the image space and predicting its position in the next video frame. The filter facilitates real time detection since the next position of the horizon can be estimated thus reducing the search area and the computational requirement of the computer vision algorithm.

Once the horizon has been detected, the orientation of the camera can be recovered using the projection model presented in Chapter 3. This measurement can be fused with an orientation measurement from an inertial unit using a quaternion extended Kalman filter to obtain the optimal estimate. In this case, the previous image domain filter becomes redundant and is superseded by the quaternion filter. The sensor fusion approach has the advantage of dealing with temporary occlusion of the horizon in the image due to the vessel roll or pitch. It relies on the result of the spatial calibration process to transform measurement from the inertial body frame to the camera body frame. Analysis of the system was presented in Chapter 5 using simulated and real data. The combination of all these algorithms make up the HoT system and is a major innovation in the work done here. Results shows that the system is capable of coping with varying sea states, illumination conditions and vessel motion.

In Chapter 6, a new algorithm for detecting cetaceans at sea was also presented. The ARCS algorithm uses the spatial and temporal properties of distinguished regions tracked in time to classify potential targets. The main contribution of the algorithm comes from the application of well-known morphological operations that do not use fixed structuring elements but instead adapt to the content of the scene. This method is adopted as an alternative to currents methods based on a fixed threshold. Results presented show that this method is capable of coping with different environmental condition without requiring any tuning of parameters.

The result of the feature extraction step is then combined with classification algorithms built from a highly imbalanced dataset. The imbalance is due to numerous clutter in the sea environment which is further exacerbated by the choice of sensor adopted here; that is, a low

cost and less sensitive thermal camera compared to those used in previous work. Also, the focus of the work here is on data from relatively warmer water which contains more sea clutter compared to images in cold water used in similar work. A data balancing scheme based on sample cleaning and k-nearest neighbour was proposed. Sample cleaning using Tomek link facilitates detecting and removing noisy samples in the larger (negative) set that appear more closely related to the opposite class, thus, reducing the risk of overfitting. A focused undersampling step is then performed using k-nearest neighbour before the training and validation of the classifier. The preferred choice of classifier used here is the SVM since it is simple and has been shown to be effective in similar work.

Experimental results presented in Chapter 6 show that the new approach is consistent following analysis using receiver-operator-characteristic curve. The benefit of data balancing as a critical step towards improving the accuracy of the algorithm was shown and the ability of the algorithm to detect features other than whale blow was also demonstrated. Results presented here prove that automated detection in warmer water is possible, albeit with a slight increase in noise due to increased sea clutter.

7.2 Future Work

The RHVM system can be improved in a number of ways as identified and outlined in this section:

Improved detection and Identification of marine species

The focus of the work here has been on the detection of whale features; other cetaceans such as dolphins have not been explored due to limited time and lack of data. Also, no real effort has been made to identify cetacean species based on their features. It is well known that certain species have uniquely identifiable features such as the “V-blow” of southern right whales and “45-degree angle blow” of the sperm whale. The review done in Chapter 2 revealed several deformable feature techniques such as active shape contours that may be suitable for this type of task.

Improved height estimation

Analysis has shown that estimation of the height of the camera above sea level is a crucial factor in the accuracy of distance estimates. Closer inspection has further revealed that height is not only affected by vessel roll and pitch but also by swell. In addition, mobile platforms such as kites and drones require a means of estimating height in real-time since it is constantly changing. A pressure sensor is the obvious first choice. However, for it to be accurate, an idea of the pressure on the sea surface must be known. It is not clear how much height estimates will be affected by using standard atmospheric pressure. An analysis is thus required, as part of future work, to investigate if it is within acceptable limits. Furthermore, alternative methods such as cheap laser range systems may also be investigated.

Fog penetration and automated determination of haze quantity

Fog remains a challenging problem for visual monitoring since both visual and thermal camera are equally affected by it. A haze removal algorithm that can alleviate this was described in Chapter 5. However, the software is not capable of automatically switching on this dehazing algorithm. It might be possible to automatically determine the quantity of haze in an image by analysing its dark channel.

The first step will be to define a decision heuristic that determines whether it is hazy. An analytical or empirical solution can be obtained by conducting experiments using image data ranging from relatively haze-free to densely-hazed images.

A camera technology that is capable of fog penetration is not known to us as at the time of writing this thesis. Short wave infrared has also been investigated and it offers no real improvements in dense fog. Development of alternative technology may also be the subject of future work.

Acceleration of computer vision algorithms using Graphical Processing Unit (GPU)

RADES and ARCS are currently accelerated using fast implementations of algorithms and the use of the several tracking filters to define a small region of interest where the horizon is detected. While the current speed is sufficient e.g. in mild or clear weather conditions, harsh weather and dehazing significantly slows it down. It has been recognised that the system can be further accelerated by taking advantage of parallel processing capabilities of a GPU.

GPUs are processors that are designed for building and rendering images to an output screen. As graphic operations become more sophisticated, more flexible GPU hardware that are user-programmable are becoming increasingly available and affordable [126]. NVIDIA have also invented CUDA, a parallel computing and programming model for their range of GPUs making them user programmable and useful in computer vision. Taking advantage of this could also be subject of future work.

References

[1] L. S. Weilgart, 'A Brief Review of Known Effects of Noise on Marine Mammals', *International Journal of Comparative Psychology*, vol. 20, no. 2, pp. 159–168, 2007.

[2] L. S. Weilgart, 'The impacts of anthropogenic ocean noise on cetaceans and implications for management', *Canadian Journal of Zoology*, vol. 85, no. 11, pp. 1091–1116, Nov. 2007.

[3] C. J. Stone and M. L. Tasker, 'The effects of seismic airguns on cetaceans in UK waters', *Journal of Cetacean Research and Management*, vol. 8, no. 3, pp. 255–263, 2006.

[4] Jonathan Gordon *et al.*, 'A Review of the Effects of Seiesmic surveys on Marine Mammals', *Marine Technology Society Journal*, vol. 37, no. 4, pp. 16–34, 2003.

[5] S. Dolman and M. Simmonds, 'Towards best environmental practice for cetacean conservation in developing Scotland's marine renewable energy', *Marine Policy*, vol. 34, no. 5, pp. 1021–1027, Sep. 2010.

[6] R. Compton, L. Goodwin, R. Handy, and V. Abbott, 'A critical examination of worldwide guidelines for minimising the disturbance to marine mammals during seismic surveys', *Marine Policy*, vol. 32, no. 3, pp. 255–262, May 2008.

[7] C. R. Weir and S. J. Dolman, 'Comparative Review of the Regional Marine Mammal Mitigation Guidelines Implemented During Industrial Seismic Surveys, and Guidance Towards a Worldwide Standard', *Journal of International Wildlife Law & Policy*, vol. 10, no. 1, pp. 1–27, Jan. 2007.

[8] E. C. M. Parsons, S. J. Dolman, M. Jasny, N. A. Rose, M. P. Simmonds, and A. J. Wright, 'A critique of the UK's JNCC seismic survey guidelines for minimising acoustic disturbance to marine mammals: best practise?', *Marine Pollution Bulletin*, vol. 58, no. 5, pp. 643–651, May 2009.

[9] D. K. Mellinger, K. M. Stafford, S. E. Moore, R. P. Dziak, and H. Matsumoto, 'An Overview of Fixed Passive Acoustic Observation Methods for Cetaceans', *Oceanography*, vol. 20, no. 4, pp. 36–45, 2007.

[10] D. Gillespie and O. P. Chappell, 'Automated cetacean detection and monitoring.', in *Proceedings of the Seismic and Marine Mammals Workshop, London.*, 1998, pp. 23–25.

[11] A. Baldacci, M. Carron, and N. Portunato, 'Infrared detection of marine mammals', *NATO Undersea Research Centre Technical Report SR-443*, 2005.

[12] Y. Podobna, J. Schoonmaker, C. Boucher, and D. Oakley, 'Optical detection of marine mammals', *SPIE Defense, Security and Sensing*, vol. 7317, p. 73170J–1–73170J–11, May 2009.

[13] Y. Podobna *et al.*, 'Airborne Multispectral Detecting System for Marine Mammal Survey', in *Proc. SPIE 7678, Ocean Sensing and Monitoring II*, vol. 7678, p. 76780G–1–76780G–9.

[14] J. Schoonmaker, J. Dirbas, Y. Podobna, T. Wells, C. Boucher, and D. Oakley, 'MultiSpectral Observation of marine Mammals', *In Proceedings of SPIE*, vol. 7113, pp. 711311–9, Oct. 2008.

[15] D. P. Zitterbart, L. Kindermann, E. Burkhardt, and O. Boebel, 'Automatic Round-the-Clock Detection of Whales for Mitigation from Underwater Noise Impacts', *PLoS one*, vol. 8, no. 8, pp. 2–7, 2013.

[16] V. Santhaseelan, S. Arigela, and V. K. Asari, 'Neural network based methodology for automatic detection of whale blows in infrared video', in *International Symposium on Visual Computing. Springer Berlin Heidelberg*, 2012, pp. 230–240.

[17] V. Santhaseelan and V. K. Asari, 'Automated Whale Blow Detection in Infrared Video', in

Computer Vision and Pattern Recognition in Environmental Informatics, 2015, pp. 58–78.

[18] J. Karnowski, C. Johnson, and E. Hutchins, 'Automated Video Surveillance for the Study of Marine Mammal Behavior and Cognition', *Animal Behavior and Cognition*, vol. 3, no. 4, pp. 255–264, 2016.

[19] R. W. Baird and S. M. Burkhart, 'Bias and variability in distance estimation on the water: implication for the management of whale watching', *International Whaling Commission SC/52/WW1*, 2000.

[20] R. Williams, R. Leaper, A. N. Zerbini, and P. S. Hammond, 'Methods for investigating measurement error in cetacean line-transect surveys', *Journal of the Marine Biological Association of the UK*, vol. 87, no. 1, p. 313, Feb. 2007.

[21] J. Gordon, 'Measuring the range to animals at sea from boats using', *Journal of Applied Ecology*, vol. 38, no. 4, pp. 879–887, 2001.

[22] D. Kinzey and T. Gerrodette, 'Distance measurements using binoculars from ships at sea: accuracy, precision and effects of refraction', *Journal of Cetacean Research and Management*, vol. 5, no. 2, pp. 159–171, 2003.

[23] Y. Hayashi, N. Wakabayashi, T. Kitahashi, and H. Wake, 'An Image Ranging System at Sea', in *Position Location and Navigation Symposium, IEEE*, 1994, pp. 3–10.

[24] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*. Prentice Hall, 2004.

[25] M. S. Nixon and A. S. Aguado, *Feature Extraction & Image Processing for Computer Vision*. Academic Press, 2012.

[26] A. A. Kassim, T. Tan, and K. H. Tan, 'A comparative study of efficient generalised Hough transform techniques', *Image and Vision Computing*, vol. 17, pp. 737–748, 1999.

[27] N. R. Harvey, R. Porter, and J. Theiler, 'Ship detection in satellite imagery using rank-order grayscale hit-or-miss transforms.', *Los Alamos National Laboratory (LANL)*, Jan. 2010.

[28] M. Khosravi and R. W. Schafer, 'Template matching based on a grayscale hit-or-miss transform', *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 1060–1066, 1996.

[29] B. Naegel, N. Passat, and C. Ronse, 'Grey-level hit-or-miss transforms — Part I : Unified theory', *Pattern Recognition*, vol. 40, no. 2, pp. 635–647, 2007.

[30] C. Barat and C. Ducottet, 'Pattern matching using morphological probing', *Image Processing, 2003. ICIP 2003. International Conference on*, vol. 1, pp. 3–6, 2003.

[31] S. Velasco-forero and J. Angulo, 'Hit-or-Miss Transform in Multivariate Images', in *Advanced Concepts for Intelligent Vision Systems*, 2010, pp. 452–463.

[32] S. Belongie, J. Malik, and J. Puzicha, 'Shape Matching and Object Recognition Using Shape Contexts', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[33] A. L. Yuille, 'Deformable Templates for Face Recognition', *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 59–70, 1990.

[34] C. Xu and J. L. Prince, 'Snakes, shapes, and gradient vector flow', *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–69, Jan. 1998.

[35] T. Cootes, 'An Introduction to Active Shape Models', *Image Processing and Analysis*, pp. 223–248, 2000.

[36] G. Hamarneh, R. Abu-gharbieh, and T. Gustavsson, 'Active Shape Models - Part I : Modeling Shape and Gray Level Variations', in *Proceedings of the Swedish Symposium on Image*

Analysis, 1998.

[37] A. Souza and J. K. Udupa, 'Automatic Landmark Selection for Active Shape Models', *In Proceedings of the SPIE*, vol. 5747, pp. 1377–1383, 2005.

[38] M. J. M. de Vasconcelos and J. M. R. da Silva Tavares, 'Methodologies to Build Automatic Point Distribution Models for Faces Represented in Images', in *Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications*, 2007, pp. 435–440.

[39] E. J. BREEN and R. JONES, 'Attribute Openings, Thinnings, and Granulometries', *Computer Vision and Image Understanding*, vol. 64, no. 3, pp. 377–389, 1996.

[40] N. Young and A. Evans, 'Psychovisually tuned attribute operators for pre-processing digital video', *IEE Proceedings-Vision, Image and Signal Processing*, vol. 150, no. 2, pp. 277–286, 2003.

[41] A. Meijster and M. Wilkinson, 'A comparison of algorithms for connected set openings and closings', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 4, pp. 484–494, 2002.

[42] C. Hsu, C. Chang, and C. Lin, 'A Practical Guide to Support Vector Classification', 2003. [Online]. Available: http://www.datascienceassn.org/sites/default/files/Practical Guide to Support Vector Classification.pdf. [Accessed: 05-Jan-2017].

[43] N. Kaempchen and K. Dietmayer, 'Data synchronization strategies for multi-sensor fusion', *In Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pp. 1–9, 2003.

[44] J. O. Nilsson and P. Handel, 'Time synchronization and temporal ordering of asynchronous sensor measurements of a multi-sensor navigation system', in *IEEE PLANS, Position Location and Navigation Symposium*, 2010, pp. 897–902.

[45] T. Kantonen, 'Sensor Synchronization for AR Applications', in *9th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2010*, 2010, pp. 245–246.

[46] E. Mair, M. Fleps, M. Suppa, and D. Burschka, 'Spatio-temporal initialization for IMU to camera registration', in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, 2011, pp. 557–564.

[47] J. Lobo and J. Dias, 'Relative Pose Calibration Between Visual and Inertial Sensors', *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 561–575, 2007.

[48] M. Li and A. I. Mourikis, '3-D motion estimation and online temporal calibration for camera-IMU systems', in *Proceedings IEEE International Conference on Robotics and Automation*, 2013, pp. 5709–5716.

[49] F. Castanedo, 'A Review of Data Fusion Techniques', *The Scientific World Journal*, 2013.

[50] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, 'Multisensor data fusion: A review of the state-of-the-art', *Information Fusion*, vol. 14, no. 1. pp. 28–44, 2013.

[51] P. Simoens *et al.*, 'Design and implementation of a hybrid remote display protocol to optimize multimedia experience on thin client devices', *Australasian Telecommunication Networks and Applications Conference*, pp. 391–396, Dec. 2008.

[52] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.

[53] Z. Zhang, 'A Flexible New Technique for Camera Calibration', in *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2000, vol. 22, no. 11, pp. 1330–1334.

[54] S. Vidas, R. Lakemond, S. Denman, C. Fookes, S. Sridharan, and T. Wark, 'A Mask-Based Approach for the Geometric Calibration of Thermal-Infrared Cameras', *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 6, pp. 1625–1635, Jun. 2012.

[55] J. Y. Bouguet, ‘Camera Calibration Toolbox for Matlab’, *Computational Vision at the California Institute of Technology*. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/. [Accessed: 14-Jan-2016].

[56] J. Sola, ‘Quaternion kinematics for the error-state KF’, 2015. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01122406/document. [Accessed: 14-Jan-2017].

[57] B. K. P. Horn, ‘Closed-form solution of absolute orientation using unit quaternions’, *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629–642, 1987.

[58] T. D. Larsen, N. a. Andersen, O. Ravn, and N. K. Poulsen, ‘Incorporation of time delayed measurements in a discrete-time Kalman filter’, in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, 1998, vol. 4, no. December, pp. 3972–3977.

[59] M. Hwangbo, J. Kim, and T. Kanade, ‘Inertial-aided KLT feature tracking for a moving camera’, in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1909–1916.

[60] A. Watt and M. Watt, *Advanced animation and rendering techniques*. ACM press, New York, USA, 1992.

[61] R. Atienza and A. Zelinsky, ‘A Practical Zoom Camera Calibration Technique: An Application of Active Vision for Human-Robot Interaction’, in *Proc. Australian Conference on Robotics and Automation*, 2001, pp. 85–90.

[62] T. Melen and J. G. Balchen, ‘Modeling and calibration of video cameras’, in *Spatial Information from Digital Photogrammetry and Computer Vision: International Society for Optics and Photonics (ISPRS) Commission III Symposium*, 1994, pp. 569–577.

[63] Z. Zhang, ‘Camera calibration with one-dimensional objects’, *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 892–899, 2004.

[64] F. C. Wu, Z. Y. Hu, and H. J. Zhu, ‘Camera calibration with moving one-dimensional objects’, *Pattern Recognition*, vol. 38, no. 5, pp. 755–765, 2005.

[65] J. Lobo and J. Dias, ‘Vision and Inertial Sensor Cooperation Using Gravity as a Vertical Reference’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1597–1608, 2003.

[66] A. T. Young and G. W. Kattawar, ‘Sunset Science. II. A useful diagram.’, *Applied optics*, vol. 37, no. 18, pp. 3785–92, Jun. 1998.

[67] T. Richardson, ‘The RFB Protocol’, *RealVNC Ltd*, pp. 1–26, 2010.

[68] Qi Baojun, Wu Tao, He Hangen, and Hu Tingbo, ‘Real-Time Detection of Small Surface Objects Using Weather Effects’, in *Asian Conference on Computer Vision (ACCV)*, 2010, pp. 27–38.

[69] N. S. Boroujeni, S. A. Etemad, and A. Whitehead, ‘Robust Horizon Detection Using Segmentation for UAV Applications’, in *Computer and Robot Vision (CRV), 2012 Ninth Conference on*, 2012, pp. 346–352.

[70] S. Fefilatyev, V. Smarodzinava, L. Hall, and D. Goldgof, ‘Horizon Detection Using Machine Learning Techniques’, in *Machine Learning and Applications, 2006. ICMLA’06. 5th International Conference on*, 2006, pp. 17–21.

[71] T. Ahmad, G. Bebis, M. Nicolescu, A. Nefian, and T. Fong, ‘An edge-less approach to horizon line detection’, in *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on*, 2015, pp. 1095–1102.

[72] L. Porzi, S. Rota Bulò, and E. Ricci, ‘A Deeply-Supervised Deconvolutional Network for Horizon Line Detection’, in *Proceedings of the 2016 ACM on Multimedia Conference*, 2016, pp. 137–141.

[73] T. Libe, E. Gershikov, and S. Kosolapov, 'Comparison of Methods for Horizon Line Detection in Sea Images', *In CONTENT 2012, The Fourth International Conference on Creative Content Technologies*, pp. 79–85, 2012.

[74] M. Schwendeman and J. Thomson, 'A horizon-tracking method for shipboard video stabilization and rectification', *Journal of Atmospheric and Oceanic Technology*, vol. 32, no. 1, pp. 164–176, 2015.

[75] D. Dusha, W. Boles, and R. Walker, 'Attitude estimation for a fixed-wing aircraft using horizon detection and optical flow', in *Proceedings - Digital Image Computing Techniques and Applications: 9th Biennial Conference of the Australian Pattern Recognition Society, DICTA 2007*, 2007, pp. 485–492.

[76] H. Yuan, X. Zhang, and Z. Feng, 'Horizon detection in foggy aerial image', in *Image Analysis and Signal Processing (IASP), 2010 international conference on*, 2010, pp. 191–194.

[77] K. He, J. Sun, and X. Tang, 'Single Image Haze Removal Using Dark Channel Prior', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, Aug. 2010.

[78] C. Zou and J. Chen, 'Recovering Depth from a Single Image Using Dark Channel Prior', in *Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), 2010 11th ACIS International Conference on*, 2010, pp. 93–96.

[79] K. He, J. Sun, and X. Tang, 'Guided Image Filtering', in *IEEE transactions on pattern analysis and machine intelligence*, 2013, vol. 35, no. 6, pp. 1397–1409.

[80] H. Xu, J. Guo, Q. Liu, and L. Ye, 'Fast image dehazing using improved dark channel prior', in *Information Science and Technology (ICIST), 2012 International Conference on*, 2012, pp. 663–667.

[81] K. Gibson, 'An investigation in dehazing compressed images and video', in *Oceans 2010 Mts/IEEE Seattle*, 2010, pp. 1–8.

[82] S. Wesolkowski, M. E. Jernigan, and R. D. Dony, 'Comparison of color image edge detectors in multiple color spaces', in *Image Processing, 2000. Proceedings. 2000 International Conference on*, 2000, vol. 2, pp. 796–799.

[83] C.-L. Liu, 'A Tutorial of the Wavelet Transform', *National Taiwan University, Department of Electrical Engineering (NTUEE), Taiwan*, 2010.

[84] K. Amolins, Y. Zhang, and P. Dare, 'Wavelet based image fusion techniques — An introduction, review and comparison', *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 62, no. 4, pp. 249–263, Sep. 2007.

[85] A. Graps, 'An introduction to wavelets', *IEEE computational science and engineering*, vol. 2, no. 2, pp. 50–61, 1992.

[86] M.-Y. Shih and D.-C. Tseng, 'A wavelet-based multiresolution edge detection and tracking', *Image and Vision Computing*, vol. 23, no. 4, pp. 441–451, Apr. 2005.

[87] S. Mallat and S. Zhong, 'Characterization of signals from multiscale edges', *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, no. 7, pp. 710–732, 1992.

[88] L. Zhang and P. Bao, 'Edge detection by scale multiplication in wavelet domain', *Pattern Recognition Letters*, vol. 23, no. 14, pp. 1771–1784, Dec. 2002.

[89] L. Feng, C. Y. Suen, Y. Y. Tang, and L. H. Yang, 'Edge Extraction of Images By Reconstruction Using Wavelet Decomposition Details At Different Resolution Levels', *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 14, no. 6, pp. 779–793, Sep. 2000.

[90] J. W. Hsieh, M. T. Ko, H. Y. M. Liao, and K. C. Fan, 'A new wavelet-based edge detector via

constrained optimization', *Image and Vision Computing*, vol. 15, no. 7, pp. 511–527, 1997.

[91] R. R. Coifman and D. L. Donoho, 'Translation-Invariant De-Noising', *Wavelets and statistics*, vol. 103, pp. 125–150, 1995.

[92] M. González-Audícana, X. Otazu, O. Fors, and a. Seco, 'Comparison between Mallat's and the "à trous" discrete wavelet transform based algorithms for the fusion of multispectral and panchromatic images', *International Journal of Remote Sensing*, vol. 26, no. 3, pp. 595–614, Feb. 2005.

[93] Y. Hao, L. Changshun, and P. Lei, 'An improved method of image edge detection based on wavelet transform', *2011 IEEE International Conference on Computer Science and Automation Engineering*, pp. 678–681, Jun. 2011.

[94] J. Canny, 'A computational approach to edge detection.', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–98, Jun. 1986.

[95] R. Medina-Carnicer, F. J. Madrid-Cuevas, a. Carmona-Poyato, and R. Muñoz-Salinas, 'On candidates selection for hysteresis thresholds in edge detection', *Pattern Recognition*, vol. 42, no. 7, pp. 1284–1296, Jul. 2009.

[96] R. Medina-Carnicer, a Carmona-Poyato, R. Muñoz-Salinas, and F. J. Madrid-Cuevas, 'Determining hysteresis thresholds for edge detection by combining the advantages and disadvantages of thresholding methods.', *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 19, no. 1, pp. 165–73, Jan. 2010.

[97] R. Medina-Carnicer, R. Muñoz-Salinas, E. Yeguas-Bolivar, and L. Diaz-Mas, 'A novel method to look for the hysteresis thresholds for the Canny edge detector', *Pattern Recognition*, vol. 44, no. 6, pp. 1201–1211, Jun. 2011.

[98] N. Otsu, 'A threshold selection method from gray-level histograms', *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[99] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, 'A new method for gray-level picture thresholding using the entropy of the histogram', *Computer vision, graphics, and image processing*, vol. 29, no. 3, pp. 273–285, 1985.

[100] W.-H. Tsai, 'Moment-preserving thresolding: A new approach', *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 377–393, 1985.

[101] P. L. Rosin, 'Unimodal thresholding', *Pattern Recognition*, vol. 34, no. 11, pp. 2083–2096, Nov. 2001.

[102] E. R. Hancock and J. Kittler, 'Adaptive estimation of hysteresis thresholds', *Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 196–201, 1991.

[103] A. L. Baruwa, 'Advanced Computer Vision System for Autonomous Underwater Vehicle: Master's thesis', University of Bath, 2010.

[104] M. Grewal and A. Andrews, *Kalman filtering: theory and practice using MATLAB*, vol. 5. Wieley-Interscience, Canada, 2001.

[105] G. Welch and G. Bishop, 'An Introduction to the Kalman Filter', *Proc of SIGGRAPH*, 2001.

[106] R. Mahony, T. Hamel, J. Pflimlin, R. Mahony, T. Hamel, and J. Pflimlin, 'Non-linear complementary filters on the special orthogonal group', *IEEE Transactions on automatic control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[107] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, 'Quaternion Averaging', *Journal of Guidance Control and Dynamics*, vol. 30, no. 4, p. 1193, 2007.

[108] J. L. Crassidis, F. L. Markley, and Y. Cheng, 'A Survey of Nonlinear Attitude Estimation

Methods', *Journal of guidance, control, and dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

[109] N. Trawny and S. I. Roumeliotis, 'Indirect Kalman Filter for 3D Attitude Estimation', *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep 2 (2005)*, 2005.

[110] D. Choukroun, I. Y. Bar-Itzhack, and Y. Oshman, 'Novel Quaternion Kalman Filter', *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 174–190, 2006.

[111] O. J. Woodman, *An Introduction to Inertial Navigation*. University of Cambridge, Computer Laboratory, 2007.

[112] W. Stockwell, 'Bias stability measurement: Allan variance', *Crossbow Technology, Inc.*, pp. 1–5, 2004.

[113] B. Mohammad and R. Mchugh, 'Automatic Detection and Characterization of Dispersive North Atlantic Right Whale Upcalls Recorded in a Shallow-Water Environment Using a Region-Based Active Contour Model', *IEEE Journal of Oceanic Engineering*, vol. 36, no. 3, pp. 431–440, 2011.

[114] J. Matas, O. Chum, M. Urban, and T. Pajdla, 'Robust Wide Baseline Stereo from Maximally Stable Extremal Regions', *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[115] D. Nistér and H. Stewénius, 'Linear time maximally stable extremal regions', in *Computer Vision–ECCV*, 2008, vol. 5303, pp. 183–196.

[116] M. Donoser and H. Bischof, 'Efficient Maximally Stable Extremal Region (MSER) Tracking', in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, vol. 1, pp. 553–560.

[117] M. Donoser, H. Riemenschneider, and H. Bischof, 'Shape guided Maximally Stable Extremal Region (MSER) tracking', in *Pattern Recognition (ICPR), 2010 20th International Conference on*, 2010, pp. 1800–1803.

[118] P. Bosilj, 'Beyond MSER : Maximally Stable Regions using Tree of Shapes', in *British Machine Vision Conference (BMVC)*, 2015, p. 169.1-169.13.

[119] L. Gomez and D. Karatzas, 'MSER-based real-time text detection and tracking', in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, 2014, pp. 3110–3115.

[120] C. Burges, 'A Tutorial on Support Vector Machines for Pattern Recognition', *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.

[121] T. Fletcher, *Support vector machines explained*. University College London, London, 2009.

[122] B. Krawczyk, 'Learning from imbalanced data: open challenges and future directions', *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

[123] H. HE and E. A. Garcia, 'Learning from Imbalanced Data Sets.', *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263--1264, 2010.

[124] I. Mani and J. Zhang, 'KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction', in *Proceedings of workshop on learning from imbalanced datasets*, 2003, vol. 126.

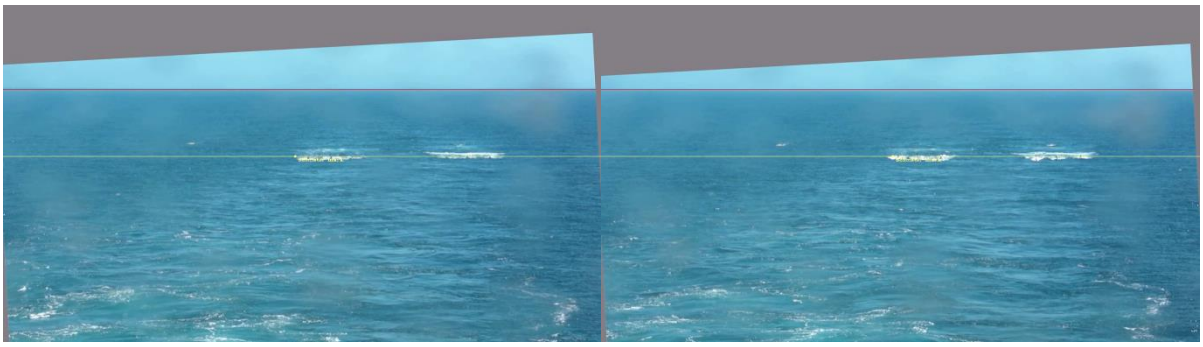
[125] V. Ganganwar, 'An overview of classification algorithms for imbalanced datasets', *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 4, pp. 42–47, 2012.

[126] J. Fung and S. Mann, 'Using graphics devices in reverse: GPU-based image processing and computer vision', in *Multimedia and Expo, 2008 IEEE International Conference on*, 2008, pp. 9–12.

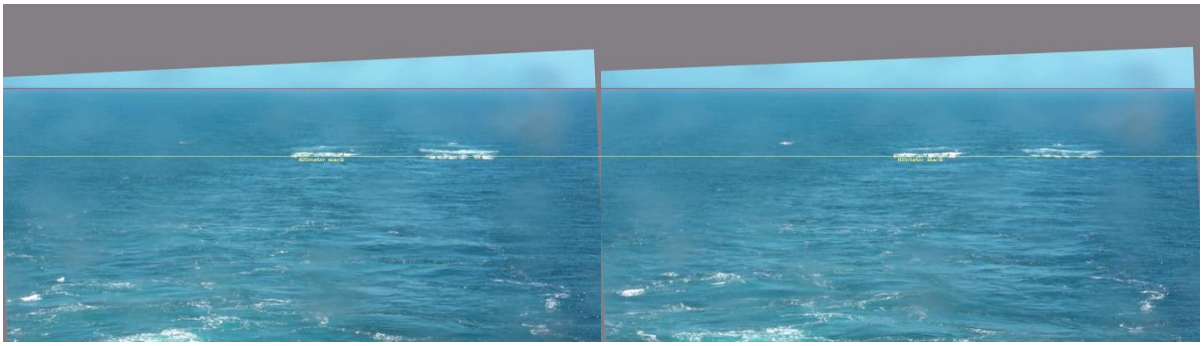
Appendix A: Further Results from Multi-Camera Sea Trials

In this appendix, additional results from trials conducted on seismic vessels offshore are shown. Sequences of images from real-time processing are presented to demonstrate horizon detection and tracking across frames as well as distance estimates. Figure A.1 shows image sequence from the verification of distance estimate using known distance to the sound source array.

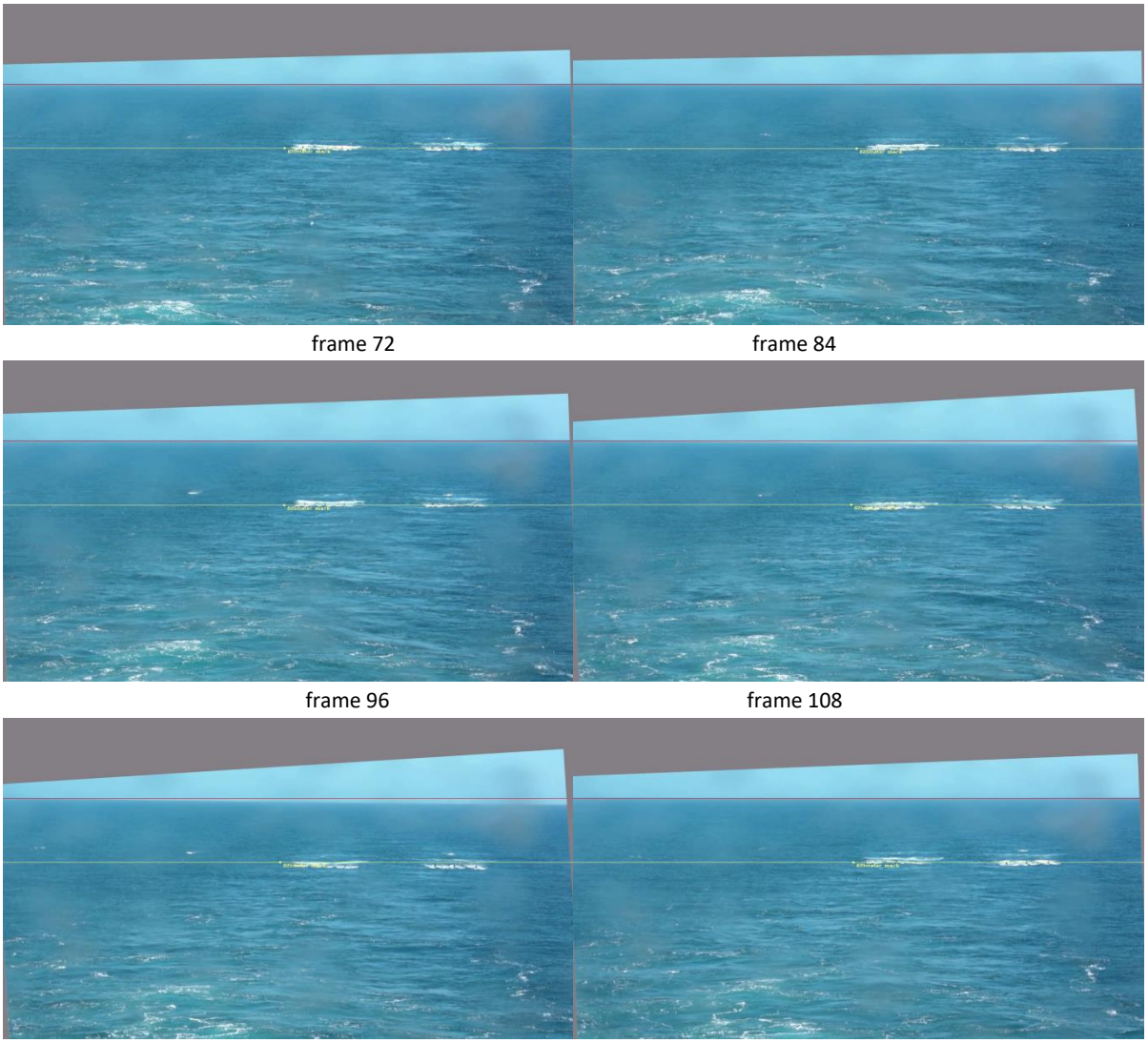
frame 0 frame 12



frame 24 frame 36



frame 48 frame 60



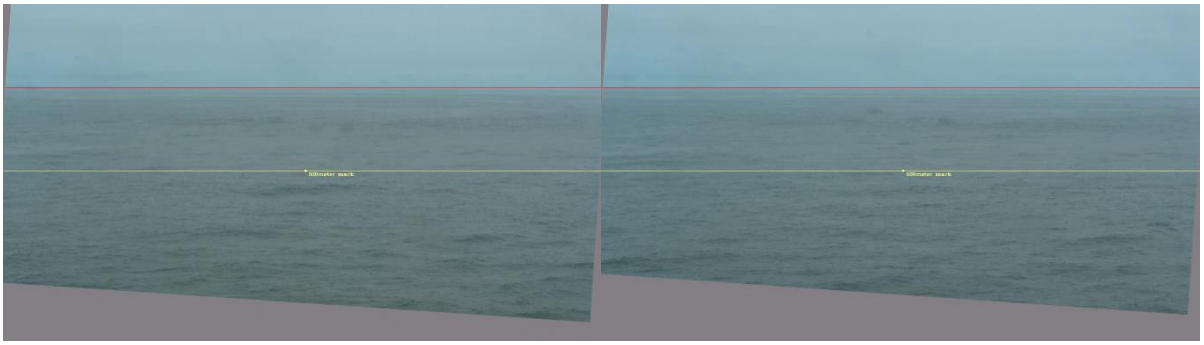
frame 72 frame 84

frame 96 frame 108

frame 120 frame 132

Figure A.1: Image sequence from verification of distance estimated by RADES to the source arrays on Camera 3

In addition to demonstrating the effectiveness of the horizon detection and tracking system across frames, the image sequence also shows the result of the image stabilisation done by the graphics engine. The grey areas in the image frame are because of the image stabilisation system. More results from with varying weather condition are shown in Figure A.2.



frame 0 frame 12

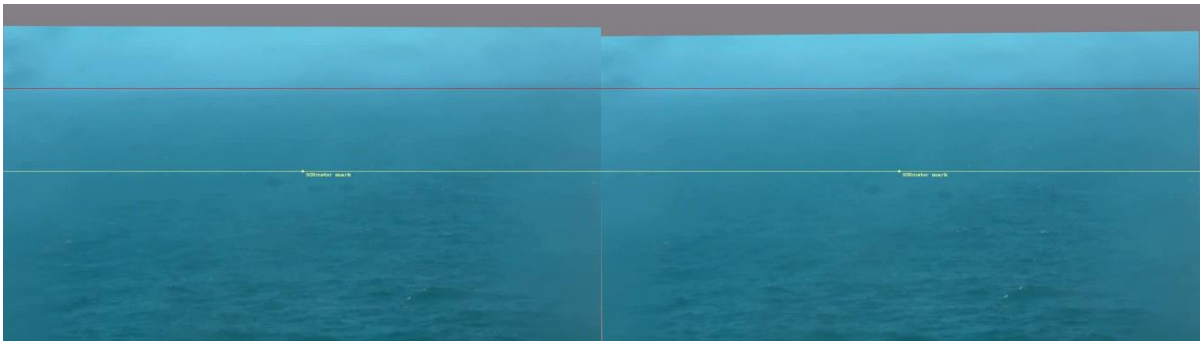
frame 24 frame 36

frame 48 frame 60

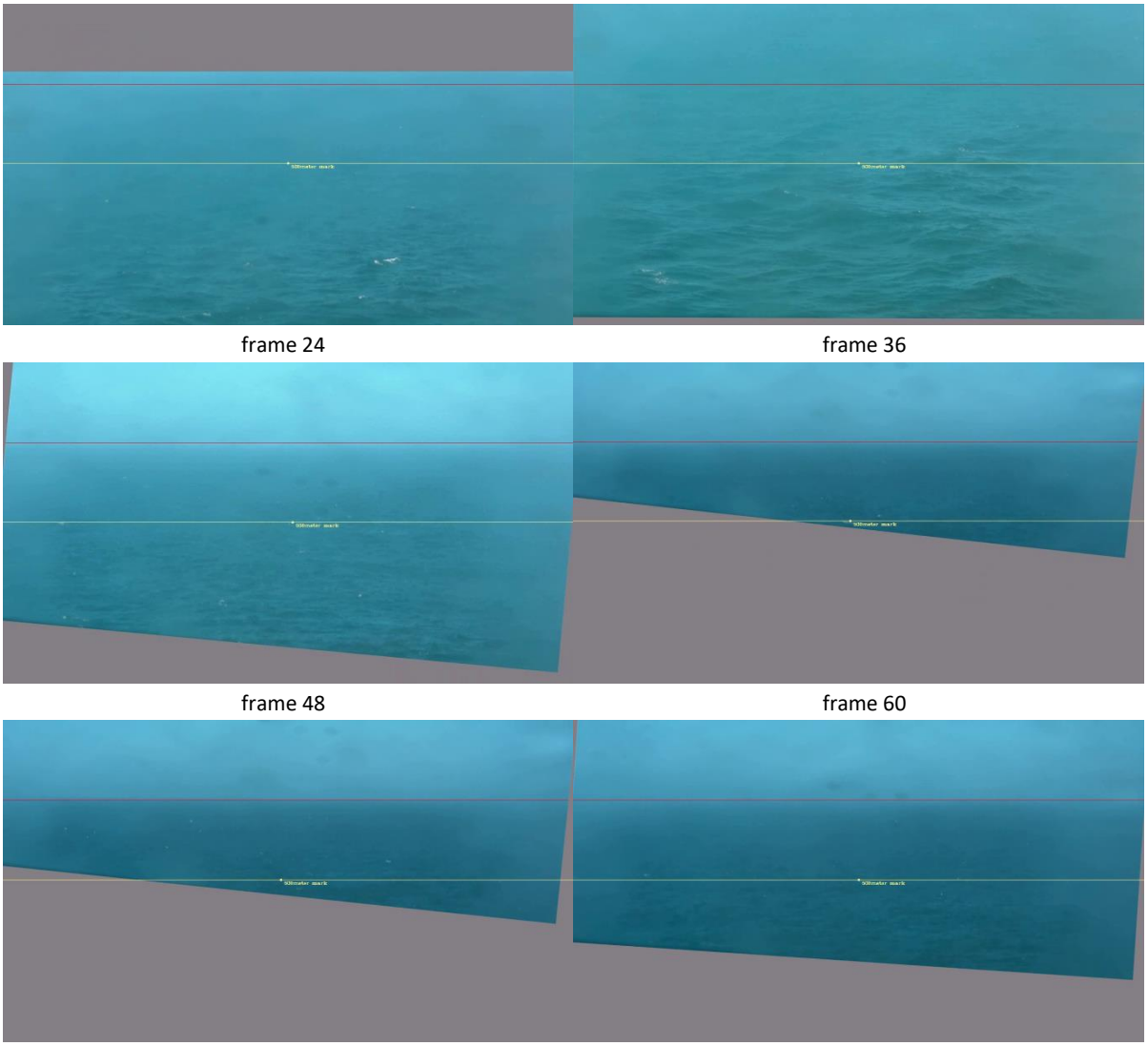
frame 72 frame 84

Figure A.2: Image sequence captured from real-time processing of images from Camera 5 on a fairly clear day

Figure A.3 shows results when it was relatively hazier and there was a lot of vessel movement. Frames 24 and 36 show the momentary failure of the horizon detection and tracking system and as a result, the failure of the image stabilisation and distance estimation system. This was due to losing the because of increased vessel motion in high sea state. Frames 48 to 84 show the recovery of the system.



frame 0 frame 12



frame 24 frame 36

frame 48 frame 60

frame 72 frame 84

Figure A.3: Image sequence captured from real-time processing of images from Camera 5 on a relatively hazy day